Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 1: Overview

Version 3.0

30 April 1996

DRAFT SF 298

1. Report Date (dd-mm-yy)
30 April 1996

2. Report Type

3. Dates covered (from... to)

4. Title & subtitle
Department of Defense Technical Architecture Framework for Information Management. Volume 1-8. Version 3.0

5a. Contract or Grant #

5b. Program Element #

6. Author(s)

5c. Project #

5d. Task #

5e. Work Unit #

7. Performing Organization Name & Address

8. Performing Organization Report #

9. Sponsoring/Monitoring Agency Name & Address
Defense Information Systems Agency
Center for Standards
10701 Parkridge Blvd
Reston, VA 20191

10. Monitor Acronym

11. Monitor Report #

12. Distribution/Availability Statement Approved for Public Release: Distribution is Unlimited

13. Supplementary Notes

14. Abstract

15. Subject Terms

16. Report
Unclass

17. Abstract
Unclass

18. This Page
Unclass

19. Limitation of Abstract

20. # of Pages

21. Responsible Person (Name and Telephone #)

Marilyn McLaughlin
(703) 735-3563

19970211 001

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 1 of the TAFIM, contains minor substantive changes from Volume 1 of Version 2.0.

Plans exist to completely revise Volume 1 to transform it to an executive summary reflecting the content of the remainder of the TAFIM. These plans could not be accomplished for Version 3.0 due to funding constraints and the volatility of a number of other TAFIM volumes.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group (AMWG) will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number changed. The next complete version will be designated Version 4.0.

- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.

- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the Defense Information Systems Agency (DISA) Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

The final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. DISA is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

CONTENTS

FIGURES

1.0 INTRODUCTION

1.1 PURPOSE

This volume presents an overview of the Technical Architecture Framework for Information Management (TAFIM). It relates information technology (IT) and information management (IM) guidance published in the Department of Defense (DoD) directives, instructions, and manuals to the TAFIM.[1]

1.2 BACKGROUND

An information system includes support and mission oriented applications, computing platforms, and communications networks. The current DoD information system technical infrastructure consists largely of stovepiped, single-purpose, and inflexible systems that are costly to maintain. These systems reflect a multiplicity of approaches to migrate toward open systems with each one progressing on its own path with limited attention to interoperability.

The evolving DoD enterprise vision for IM emphasizes integration, interoperability, flexibility, and efficiency through the development of a common, multi-purpose, standards-based technical infrastructure. This vision requires a new paradigm for building technical architectures and information systems that improve the effectiveness of functional operations to include their efficiency and use of technology throughout the DoD.

The emerging concepts for warfighting depend upon information being managed as a Department-wide resource. Joint campaigns should fully exploit the "information differential," which is the superior access to and ability to effectively employ information on the strategic, operational, and tactical situation that advanced United States (U.S.) technologies can provide our forces. This information differential requires a seamless interface between the "foxhole" and the support base, between intelligence and operations, and between the DoD and its suppliers. However, today there is no unifying DoD IM technical architecture guidance that can satisfy these goals.

In the absence of DoD-wide IM technical architecture guidance, the Services, Agencies, and Commanders-in-Chief (CINCs) have developed a wide range of architectures to manage and control their technical infrastructures. Reference models, information architectures, communications architectures, mission architectures, and various other architectures are now used to manage the design and development of technical infrastructures and information systems within the Services, Agencies, and CINCs.

[1] A list of references is contained in Appendix A. Reference 1 identifies the Executive Level Guidance, which is the source for the IT vision in Section 3 and the IM vision in Appendix C. References 2 through 9 are DoD directives, instructions, and manuals, all of which directly relate to the TAFIM. Reference 10 contains guidance for the preparation of Functional Economic Analyses.

The Technical Reference Model (TRM) for IM was the initial effort to bring commonality and standardization to the technical infrastructure. The TRM addresses the services and standards needed to implement a common technical infrastructure. A single technical architecture framework was needed to integrate these efforts and drive systems design, acquisition, and reuse throughout the DoD.

The single technical architecture framework is the TAFIM. It provides the DoD-wide framework to manage multiple technical architecture initiatives. It is intended to achieve the following results:

- The use of common principles, assumptions, and terminology in the DoD Component (Services, Agencies, and CINCs) technical architectures
- The definition of a single structure for the DoD technical infrastructure components (system components) and how they are managed
- The development of information systems in accordance with common principles to permit DoD-wide integration and interoperability.

1.3 TAFIM PURPOSE

The TAFIM provides guidance for the evolution of the DoD technical infrastructure. The TAFIM does not provide a specific system architecture. Rather, it provides the services, standards, design concepts, components, and configurations that can be used to guide the development of technical architectures that meet specific mission requirements.

The TAFIM is independent of mission-specific applications and their associated data. It introduces and promotes interoperability, portability, and scalability of DoD information systems. The TAFIM is an Enterprise Level[2] guide for developing technical architectures that satisfy specific functional requirements. It also provides an organizational level guide and link to the Enterprise Level. To achieve an integrated enterprise, it is assumed that all information systems must interoperate at some time. Therefore, their architects and designers should use the TAFIM as the basis for developing a common target architecture to which systems can migrate, evolve, and interoperate. Over time, interoperability between and among the number of systems will increase, providing users with improved services needed to achieve common functional objectives. To achieve portability, standard interfaces will be developed and implemented. Scalability will be developed in mission applications to accommodate flexibility in the functionality. Proper application of the TAFIM guidance can:

- Promote integration, interoperability, modularity, and flexibility

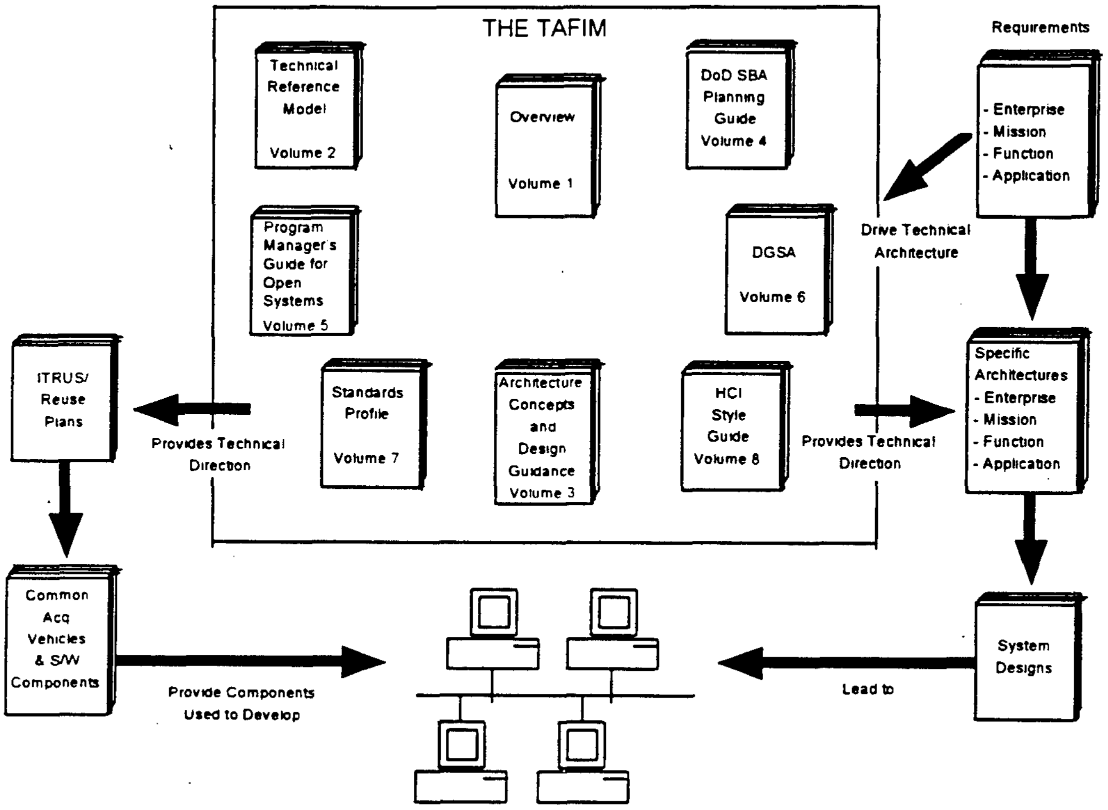
[2] This should be read as Departmental- or DoD-Level, which are synonymous with Enterprise Level.

- Guide acquisition and reuse

- Speed delivery of information technology and lower its costs.

1.4 SCOPE AND APPLICABILITY

The TAFIM applies to information system technical architectures at all DoD organization levels and environments (e.g., tactical, strategic, sustaining base, interfaces to weapons systems) – see Appendix D for further guidance regarding applicability. As Figure 1-1 shows, the TAFIM is intended to guide the development of architectures that satisfy requirements across missions, functional areas, and functional activities [DoD 8020.1-M]. The TAFIM is mandatory for use in DoD. The specific technical architectures for missions and functions will be developed using standard architecture guidance and development methodologies provided by the TAFIM.



THE TAFIM
Technical Reference Model Volume 2
Overview Volume 1
DoD SBA Planning Guide Volume 4
Program Manager's Guide for Open Systems Volume 5
DGSA Volume 6
Standards Profile Volume 7
Architecture Concepts and Design Guidance Volume 3
HCI Style Guide Volume 8
Requirements
- Enterprise
- Mission
- Function
- Application
Drive Technical Architecture
Specific Architectures
- Enterprise
- Mission
- Function
- Application
Provides Technical Direction
Provides Technical Direction
ITRUS/ Reuse Plans
Common Acq Vehicles & S/W Components
Provide Components Used to Develop
System Designs
Lead to

Figure 1-1. Architecture Implementation Concept

1.5 DOCUMENT ORGANIZATION

Section 2 describes the TAFIM structure and content. Section 3 presents the DoD vision for information technology. Sections 4 and 5 address the information system life cycle and IM integration model, respectively. Appendix A is a list of references. Appendix B defines acronyms and provides a glossary of terms used in the TAFIM. Appendix C provides the DoD vision for IM. Appendix D is the text of three DoD memoranda that provide guidance for using the TAFIM in developing technical architectures. Appendix E provides a format and guidance for proposing changes to this document.

2.0 TAFIM DESCRIPTION

2.1 INFORMATION SYSTEMS

An information system (IS) consists of mission-specific applications, data, and technical infrastructure architecture consisting of support applications, application platforms, and the external environment including devices such as terminals, printers, and communications networks. Each of these elements has a unique life cycle that requires distinct development and maintenance approaches. For example, data definitions and formats may have a useful life that is many times longer than the mission-specific applications that manipulate and use the data definitions, and the hardware and software that comprise the technical infrastructure architecture may have a life half as long as the mission-specific applications. Each of these elements should be managed according to its life cycle. An information system architecture (ISA) is presented in Figure 2-1 and shows a physical separation of the elements and reflects a mission-specific application software architecture, a data architecture, and a technical infrastructure architecture, which is sometimes referred to as the technical infrastructure architecture.

The data architecture supports standard data elements, data integrity, data availability, shared databases, and the separation of applications and data. The application software architecture supports the development of reusable applications, which are independent of data and the platforms on which they run. The technical infrastructure architecture describes the support applications, computing platforms including the operating system, and external environment needed to provide the connectivity or interoperability for applications and data.

2.2 THE TAFIM VOLUMES

The TAFIM provides a set of volumes for guiding the evolution of the DoD's technical architecture, which consists of multiple environments with each environment accommodating one or more ISAs. The TAFIM consists of multiple volumes in various states of development and maturity.

The volumes that constitute Version 3.0 of the TAFIM are listed below.

- Volume 1: *Overview* (this document).

- Volume 2: *Technical Reference Model* provides the conceptual model for information system services and their interfaces.

- Volume 3: *Architecture Concepts and Design Guidance* provides concepts and guidance needed to support the development of technical architectures in the DoD.

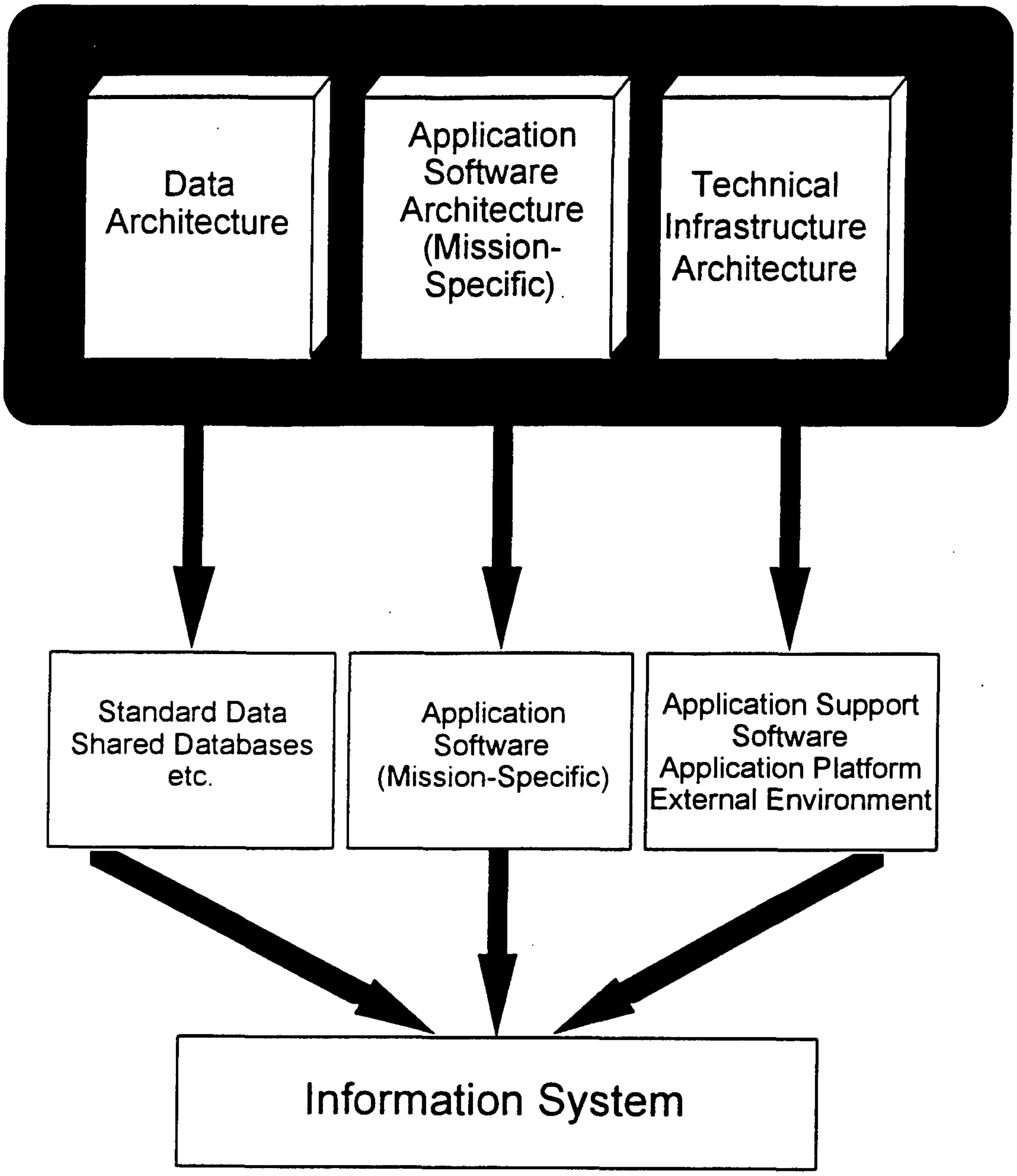

Data Architecture
Application Software Architecture (Mission-Specific).
Technical Infrastructure Architecture
Standard Data Shared Databases etc.
Application Software (Mission-Specific)
Application Support Software Application Platform External Environment
Information System

Figure 2-1. Information Systems Architecture

- Volume 4: *DoD Standards-Based Architecture Planning Guide* provides a standards-based architecture planning methodology that will help architects, technical integrators, and developers to plan and build information systems that meet mission, functional, and application area requirements. The methodology provides a translation of functional requirements to the selection of services, standards, components, configurations, their phasing, and the acquisition of products that implement them.

- Volume 5: *Program Managers Guide for Open Systems* describes how to use the TAFIM guidance in the acquisition of IT and IM products.

- Volume 6: *DoD Goal Security Architecture* (DGSA) addresses security requirements commonly found within DoD organizations' missions or derived as a result of examining mission threats. Further, the DGSA provides a general statement about a common collection of security services and mechanisms that an information system might offer through its generic components. The DGSA also specifies principles, concepts, functions, and services that target security capabilities to guide system architects in developing their specific architectures. The generic security architecture provides an initial allocation of security services and functions and begins to define the types of components and security mechanisms that are available to implement security services. In addition, examples are provided of how to use the DGSA in developing mission-level technical architectures.

- Volume 7: *Adopted Information Technology Standards* (AITS) is the definitive set of IT standards to be used in DoD. It is intended to guide DoD acquisitions and the migration of legacy systems and, by providing definitive standards, to support broader TAFIM objectives such as interoperability, reduced life-cycle costs, and security.

- Volume 8: *DoD Human Computer Interface (HCI) Style Guide* provides a common framework for HCI design and implementation.

This page intentionally left blank.

3.0 THE VISION FOR DOD INFORMATION TECHNOLOGY

This section focuses on the vision [Executive Level Guidance (ELG)] for DoD information technology. It is part of the total DoD guidance for planning, developing, and operating the DoD's information systems. Implementing state-of-the-art information technology provides for improved information management. The TAFIM furthers this concept. It also supports the information management vision, described in Appendix C. They both relate to the DoD information systems technical infrastructure.

Information technology is integral to providing efficient and effective functional information management processes and practices across the DoD. It is recognized as a force multiplier during peacetime, transition to war, and war. The implementation of information technological principles and products into all aspects of DoD operations means that effective military capabilities can be maintained within smaller defense budgets.

3.1 TECHNOLOGY

Off-the-shelf information technology is becoming more flexible and powerful. Within DoD, this information technology eventually will extend from the foxhole to the office, in fixed and mobile locations, across the full spectrum of peace, transition to war, and war. It will be ubiquitous and integral to all DoD operations and user tasks.

The information technology will make possible capabilities that encompass all composite objects consisting of different types of related temporal and logical content that can be entered, accessed, manipulated, and displayed at every workstation as an integral part of each job. Workstation platforms and other user devices that become available in the early twenty-first century are expected to be many times more powerful than the machines of the early 1990s. Workstations will adhere to a full suite of Federal, national, and international standards that have been adopted by the DoD. Because platforms adhere to a common set of interface standards, it will be possible to configure software across a distributed environment and tailor the software to support specific functional processes. The ubiquity of standard low-cost platforms, coupled with rapid and responsive software development, will enable effective implementation of continuous functional process improvements.

3.2 PRODUCT AVAILABILITY

Commercial software products, supplemented (when necessary) by Government-developed reusable components, will provide DoD's IM system developers with powerful tools to enhance productivity and decision making. The accumulated experience of DoD personnel will be preserved through standard databases that are portable across platforms, locations, applications, and assignments. Users also will be provided with the tools to tailor screens, menus, and applications so that they can be more productive, innovative, and effective in the performance of assigned duties. Policies, procedures, standards, and controls will govern this individual capability, ensuring that its use is consistent with military doctrine and mission IM standards.

3.3 ROUTINE OPERATIONS

DoD information systems and their associated improved processes will perform many of the current individual manual and routine operations, allowing individuals to perform value-added work. With such capabilities, individuals and groups may dynamically configure information resources (e.g., data, processing resources). In effect, users will set up their own virtual operations/work spaces and use them to get the immediate task accomplished. When a task is finished, the resources will be returned to a common pool, and new tasks will begin. This reconfigurable information resources model enables developers to create an environment that supports routine work as well as serving dynamic battle situations with technology that transitions smoothly from peace to war.

3.4 OPEN SYSTEMS ENVIRONMENT

DoD is fully committed to implementing an open systems environment (OSE). This environment will enable information systems to be developed, operated, and maintained independent of application-specific technical solutions or vendor products. DoD is establishing a standards-based framework for defining technical architectures to provide interoperability, portability, and scalability. System attributes such as performance, response time, and availability, which are not part of the open system, will be separately defined within the requirements of the functionality as implemented in each Automated Information System (AIS). The TAFIM uses Federal and national standards adopted by industry, and international standards accepted worldwide by U.S. allies. The guidelines will show technical managers and developers at all levels of the DoD how to create profiles of standards to meet specific mission-area architecture needs. Also, the guidelines will provide transition strategies on how to evolve baselines and legacy systems to the target open environment. When developing information systems, the DoD Components and subordinate commands will follow the guidelines and apply the standards recommended by TAFIM. This will enable all functions to work together, and all systems to benefit from the efficiencies made possible through the shared part of the DoD infrastructure.

DoD has and will continue to play a leadership role in the development of standards that contribute to open systems by working in concert with national, international, and industry bodies. DoD is beginning to work with vendors to ensure they incorporate standards recommended by TAFIM, capabilities, and features in their products for use in DoD systems.

3.5 DATA AND INFORMATION SECURITY

Security of vital DoD information resources will be achieved through a common approach to integrated policy, architecture, and engineering using the DGSA concepts in conjunction with other DoD guidance. Security architectures will satisfy mission-area security policies and align with TAFIM-recommended standards that address open systems. The protection of information and system assets will be part of the total security requirement for automated services. DoD systems support information processing under arbitrarily complex security policies, including those involving support of multiple categories of sensitive classified and unclassified information. The systems will be sufficiently protected to allow distributed processing among multiple hosts on multiple networks in accordance with open system architectures. They support information processing among users employing resources with various types of security protection, including users of non-secure resources if a particular mission so dictates. The DoD information systems will be sufficiently protected to allow connectivity via common carrier (public) communication systems.

The DGSA will allow different mission-area information systems to exchange information in a secure manner yet ensure the integrity, confidentiality, availability, and authenticity of enterprise databases and resources.

3.6 THE DOD INFORMATION UTILITY

The DoD will operate an information utility that users can access worldwide to obtain needed information services. The information utility will be transparent and will deliver a full spectrum of quality services, where and when needed, tailored to the job, affordably priced to match alternative sources, when appropriate and available. This environment will be managed from a DoD-wide perspective to achieve a balance of centralized, local, and individual capabilities. All DoD shared information resources, both owned and leased, form a global network that will be centrally managed as part of the overall systems and networks of the Defense Information Infrastructure (DII) across the various environments, including:

- Central processing centers that house the master copies of corporate databases and perform large-scale production jobs

- Fixed site installations and mobile facilities where application processing occurs, where networks and systems are managed, and where the data are captured and stored for local use

- The personal computing environments that enable individuals to manage their information resources.

3.7 SHARED DATABASES

Shared databases will be established, centrally managed, and controlled to ensure the integrity of the information resource for the entire DoD. Rules and mechanisms will be put in place to allow individuals to make individual use of data while maintaining the data standards established for all users, including appropriate security controls. Data that crosses DoD Component or functional boundaries will be kept in shared databases and accessed over the common-user global network. These corporate-type databases will be governed by consistent data models, centrally managed, logically integrated, and physically distributed worldwide, with automated backup and recovery. The DGSA is an integral part of the TAFIM. It specifies security principles and targets security capabilities that will guide system architects in creating specific architectures that will meet mission security policies.

3.8 BACKBONE NETWORK

The DoD will establish, operate, and centrally manage a Defense Information System Network (DISN) as part of the DII that will evolve to make use of highly available, ubiquitous, global, commercial communications networks for the vast majority of the DoD communications needs. These networks will feature the cost savings of bandwidth-on-demand service and integrated services for voice, data, and video applications. The DISN will provide value-added services for secure and non-secure directories, conferencing, and databases. The DISN will also provide backbone connectivity between users who require the special protection of complete traffic flow security.

This backbone connectivity will eventually extend to desktops and mobile devices. It will be survivable, robust, and centrally managed to optimize the use of resources, availability, and performance. A security architecture, using DGSA concepts, and new procedures will allow different functional communities to exchange information easily while maintaining the integrity of their mission areas.

3.9 STREAMLINED LIFE CYCLE

A streamlined life cycle will be used to compress the time needed to deliver new capabilities to the field and to reduce total life-cycle costs. The process will emphasize the use of powerful and integrated computer-assisted methodologies and tools such as the shared utility services, reuse of software components, refurbishment and replenishment of hardware acquired as a commodity item, building-block construction of systems, use of products meeting the DoD architecture guidelines and standards, and improved technical management. Ad hoc system development efforts will not be permitted. System developments will be organized and engineered to be repeatable and reliable so as to achieve quality, efficient, and effective rapid production.

3.10 MODELING AND PROTOTYPING

Data modeling is becoming mature. It will be fully integrated with process modeling in a common DoD-wide approach. Powerful and integrated computer-assisted development and maintenance environments will rapidly capture process models, data models, and other requirements and transform them into applications and databases that adhere to DoD standards for data elements and software. Rapid prototyping will be a built-in aspect of the systems development cycle, so that incremental changes that support improved business processes can be accomplished in days and weeks rather than months and years.

3.11 STREAMLINED ACQUISITION

A streamlined acquisition process will be functioning that ensures the implementation of the DoD information system infrastructure can be achieved on schedule and within budget. Compliant components will be available from "one-stop shopping" technology "stores" when they are needed. Hardware and most generic software components (e.g., database management systems, electronic mail (E-mail) packages) will be acquired as products that serve mission-area applications, which embody specific business rules and user interactions.

Acquisition lead-times will be shortened to avail the DoD of new cost-effective technology's best suite to improve functional processes. Open system standards will expedite the acquisition process by reducing the time and cost of migrating to improved environments. Innovative mechanisms, such as hardware leasing, will be in place to acquire a full spectrum of information products and services at the best cost value to the Government. Products may be procured as new, reused, or refurbished in a cost-effective manner. These improvements will be supported by test and evaluation (T&E) methodologies that are being overhauled to support the rapid acquisition of information systems.

3.12 PERFORMANCE

The DoD technical infrastructure will be founded on a baseline of standard configurations that will provide the required performance within cost. Measures of effectiveness (MOE) will be used to evaluate how well the infrastructure is supporting the functional users. The application of MOEs (including benchmarking against industry best practices) will assure DoD managers that the infrastructure technology is effective and efficient and that the service provided compares favorably with the commercial support provided to the public sector. IT will be managed, in the same way as other IM activities are managed, to enable continual improvement. Although IM has to be managed, in an authoritarian organization like DoD, use of open systems assumes that the end users (action officers, not clerks) have a wide range of tools, capabilities, and applications with appropriate access to enterprise data. Once this is granted, the users will be empowered and authorized to utilize this information technology. The end use of the system should not be managed – rather, the effectiveness of providing that environment to the users should be managed.

3.13 EDUCATION AND TRAINING

Education and training of the DoD IM community in new methods, tools, and practices will be centrally managed. The goal will be to create technically literate users, who can obtain the maximum benefits from the new technologies. There will be a renewed emphasis on enhancing individual skills, productivity, professional growth, and job satisfaction. This emphasis recognizes that DoD personnel are the most important DoD resource.

This page intentionally left blank.

4.0 INFORMATION SYSTEM LIFE-CYCLE SUPPORT

4.1 LIFE-CYCLE MANAGEMENT

The TAFIM supports life-cycle management (LCM) as published in DoD guidance directives [DoD Directive (DoDD) 8120.1 and DoD Instruction (DoDI) 8120.2]. It also supports the LCM method of reporting system development progress to decision makers and specifically addresses those efforts that take place in the development phase of new information systems or in the update to existing information systems.

4.2 INFORMATION SYSTEMS DEVELOPMENT

The TAFIM supports evolutionary, incremental, and concurrent development methods that contribute to reducing the time it takes to field new or revised capabilities. Whatever method is selected, it is documented in life cycle documentation presented to decision makers for approval. Figure 4-1 presents a method where requirements are identified as input to the development and operation of an information system.

The figure relates TAFIM guidance, development aids, tools, and products to the development cycle. The developer should take every advantage of the TAFIM guidance and of available development tools and aids. Development support includes prototyping, standardized data and database sharing, procuring commercial-off-the-shelf (COTS) products, reusing common applications software, implementing common-use infrastructure services (computer and communications utility), and using integrated computer-aided software engineering (I-CASE) tools. The products and services are standards-based and architecturally driven. The use of standards and common technical architectures will reduce the likelihood that stove-pipe systems will be developed. This should result in system components that are interoperable, compatible, flexible, and operationally efficient, even though they are acquired and configured by different executive agents.

Within common architectures, applications, data, and infrastructures must be managed according to their separate life cycles. To make this approach work, the various support tools and mechanisms for designing, prototyping, developing, acquiring, integrating, testing, fielding, and operating information systems must adhere to the common architecture principles, guidelines, and standards. Their implementation should employ innovative methods, tailored to meet the situation associated with the requirements. The blocks shown in Figure 4-1 are briefly discussed in the following subsections.

4.2.1 Requirements Definition

The Enterprise Model described in DoD 8020.1-M provides the framework for developing integrated process and data models for specific functional activities in the DoD. Together, these

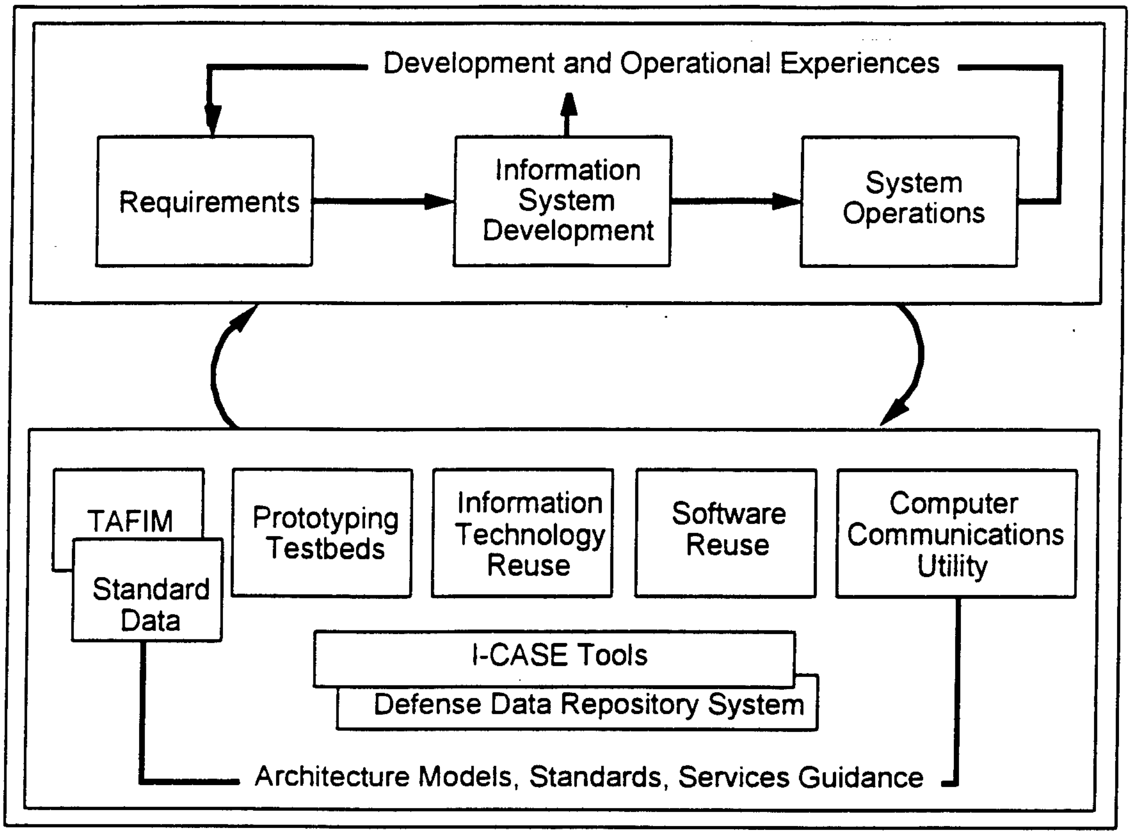

Development and Operational Experiences
Requirements
Information System Development
System Operations
TAFIM
Standard Data
Prototyping Testbeds
Information Technology Reuse
Software Reuse
Computer Communications Utility
I-CASE Tools
Defense Data Repository System
Architecture Models, Standards, Services Guidance

Figure 4-1. Information Systems Life-Cycle Support

models specify the functional user (logical) requirements for an information system. In addition to addressing the foregoing, DoD 8020.1-M addresses the DoD Data Administration Strategic Plan (DASP) and other DoD IM documents. The model requirements provide input for developing the technical architecture addressed in the TAFIM. The requirements are established using a DoD standard methodology, described in Chapter 8, DoD 8020.1-M. This or other methodologies provide the requirements input for information system development.

4.2.2 Information Systems Development

The TAFIM provides guidance to architects and designers on the selection of compatible configurations of standards, services, and components that can be implemented through common-use acquisitions, DoD software reuse libraries, and shared utility services (e.g., a global network). Development activities define an ISA that is based on functional requirements and consists of the data architecture, application architecture, and technical architecture. The technical architecture guidance is provided by the TAFIM. The data and mission application software architectures [DoDDs 4630.5, 8000.1, 8120.1, 8320.1] are developed by mission or function. Together they require integration into the overall infrastructure.

To assist the development activity, the TAFIM includes a reference model and services, a tailorable standards profile, architecture concepts, and design guidance. Information system development efforts include rapid design and prototyping. These efforts include the use of corporate data, reusable software, and infrastructure "building blocks" from various DoD IM initiatives that are being documented in the TAFIM. Detailed engineering guidance, particularly for migrating from or interfacing to legacy environments, is outside the current scope of the TAFIM.

TAFIM Volume 4, *DoD Standards-Based Architecture Planning Guide*, provides a standards-based architecture development methodology. In general, this methodology starts with the functional models and requirements and includes evaluating the baseline for deficiencies and opportunities, selecting a target or open architecture, and identifying migration paths and actions to evolve from the baseline to the target architecture. This process involves integrating the data architecture, mission application architecture, and technical architecture into a total ISA.

In support of the TAFIM, the Defense Data Repository System (DDRS) will be integrated with I-CASE, the IDEF repository, and the software reuse libraries. The DoD Software Reuse Program will provide software components that implement standards recommended by TAFIM and its guidance. An example would be software modules that use standard application program interfaces (APIs). Applications developed by specific functional communities will be put in central libraries and made available to development activities. The concept allows for lead development activities that develop integrated sets of application software for functional domains, including shared system software. Software components developed according to Software Reuse Program standards and design guidelines must be consistent with the TAFIM to promote reuse, portability, and interoperability of systems in the DoD. I-CASE tools and integrated software development methods will be selected and configured to support the TAFIM. For example, I-CASE tools will generate code that uses the APIs specified in the TAFIM.

Prototyping environments will adhere to the TAFIM guidelines and standards and use information technology reuse (ITRUS) components, DoD software reuse products, and I-CASE prototyping tools to the maximum extent possible. This will facilitate rapid prototyping of applications and databases that can be validated by users and easily transitioned into production environments.

4.2.3 System Operations

Information systems will be operated in the global computer and communications utility environment that adheres to standards recommended by TAFIM and its guidelines. This will promote portability, survivability, flexibility, and interoperability for all DoD information systems. Centrally managed processing centers, global networks, sustaining base installations, and tactical environments will be developed using the basic approach outlined above. Databases and applications that use the standards recommended by TAFIM and its design features will become largely independent of where they are hosted. They will be easily portable across the infrastructure environment, allowing efficient resource utilization, backup, and least-cost utility service to the customer.

4.3 INFORMATION SYSTEMS EVOLUTION

The TAFIM provides the basis for interoperability of information systems by defining common services, standards, and configurations for the DoD technical infrastructure (i.e., support applications, application platforms, and communications networks). New DoD information systems will achieve interoperability by being built in conformance with an ISA based on the design guidance and standards set forth in the TAFIM. Interoperability of existing systems will be increased by evolving them to ISAs that are consistent with the TAFIM.

To evolve existing systems, functional and technical teams assess existing systems as part of the mission-area or DoD Component-wide strategic planning process. These teams determine the degree that the existing systems are in compliance with functional requirements and provide required services. They also assess how well existing systems meet standards that accommodate open systems. These teams determine and evaluate the cost, time, and risk required to evolve existing systems to the goal architecture. These assessments can be an input to the Functional Economic Analysis (FEA) [DoD Corporate Information Management (CIM)] that is a consideration in the process of selecting existing systems for migration or authorizing a new start AIS.

The rate at which different system baselines converge to the open systems architecture is governed by many factors, including the need to select migration systems and to develop them to a common open architecture in the DoD, and in so doing, implement functional process improvements. Many systems are currently implemented in unique or proprietary environments from which it is difficult to evolve. Figure 4-2 shows how migration systems and other systems will go through several phases in their convergence to an open systems target architecture during the 1990s and beyond.

The first phase is constrained by the need to continue some legacy systems while selecting others as standard migration systems. Therefore, near-term target architectures will continue to have legacy and proprietary elements that must interface with migration systems as they evolve to open systems elements. Once the target baseline is achieved, there will be greater opportunities to satisfy functional process improvement support needs with open systems solutions. Finally, systems can be planned so as to evolve to standards that accommodate open systems.

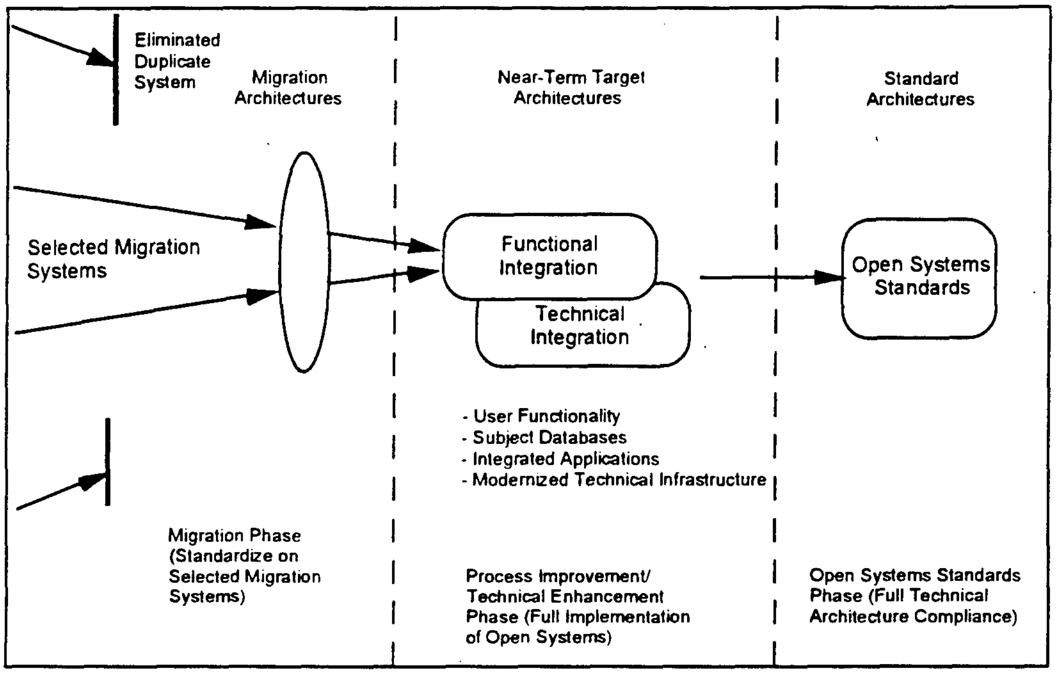

Eliminated Duplicate System
Migration Architectures
Near-Term Target Architectures
Standard Architectures
Selected Migration Systems
Functional Integration
Technical Integration
Open Systems Standards
- User Functionality
- Subject Databases
- Integrated Applications
- Modernized Technical Infrastructure
Migration Phase (Standardize on Selected Migration Systems)
Process Improvement/ Technical Enhancement Phase (Full Implementation of Open Systems)
Open Systems Standards Phase (Full Technical Architecture Compliance)

Figure 4-2. Phased Convergence To DoD Open Systems Architecture

This page intentionally left blank.

5.0 INFORMATION MANAGEMENT INTEGRATION MODEL

5.1 INTRODUCTION

Functional and technical integration of user requirements presents significant potential for cost savings and system flexibility. Since, user requirements differ in a number of ways, their integration can mean that the user will not require multiple products or services to meet these multiple needs.

5.2 OBJECTIVE

The objective of integration [DoD 4630.5 and DoDI 4630.8] is to:

- Achieve or improve system interoperability
- Achieve compliance with international, national, and DoD open systems standards
- Provide users a single common interface
- Achieve portability and flexibility.

5.3 DESCRIPTION OF THE INTEGRATION MODEL

Integrating functional and technical requirements of DoD information systems can be portrayed using the DoD IM integration model shown in Figure 5-1. It represents a perspective for defining boundaries for potential integration pay-off within DoD IM activities from a DoD-wide view. Further, it can assist integrators in defining what is to be integrated in order to correctly proceed with the task. Functional and technical integration requirements must be addressed both at the vertical boundaries within a level and the horizontal boundaries between the levels of the model.

5.4 TYPES AND LEVELS OF INTEGRATION

Integration can occur within or between the levels of the model but the requirements for the type of integration must still be defined. To gather these detailed requirements, significant research and analysis efforts may be required to gain a full understanding of the integration task. Integration should result in interoperability and efficiency, effectiveness, optimization, resource savings, or other benefits. Integration will be viewed from at least one of the following perspectives:

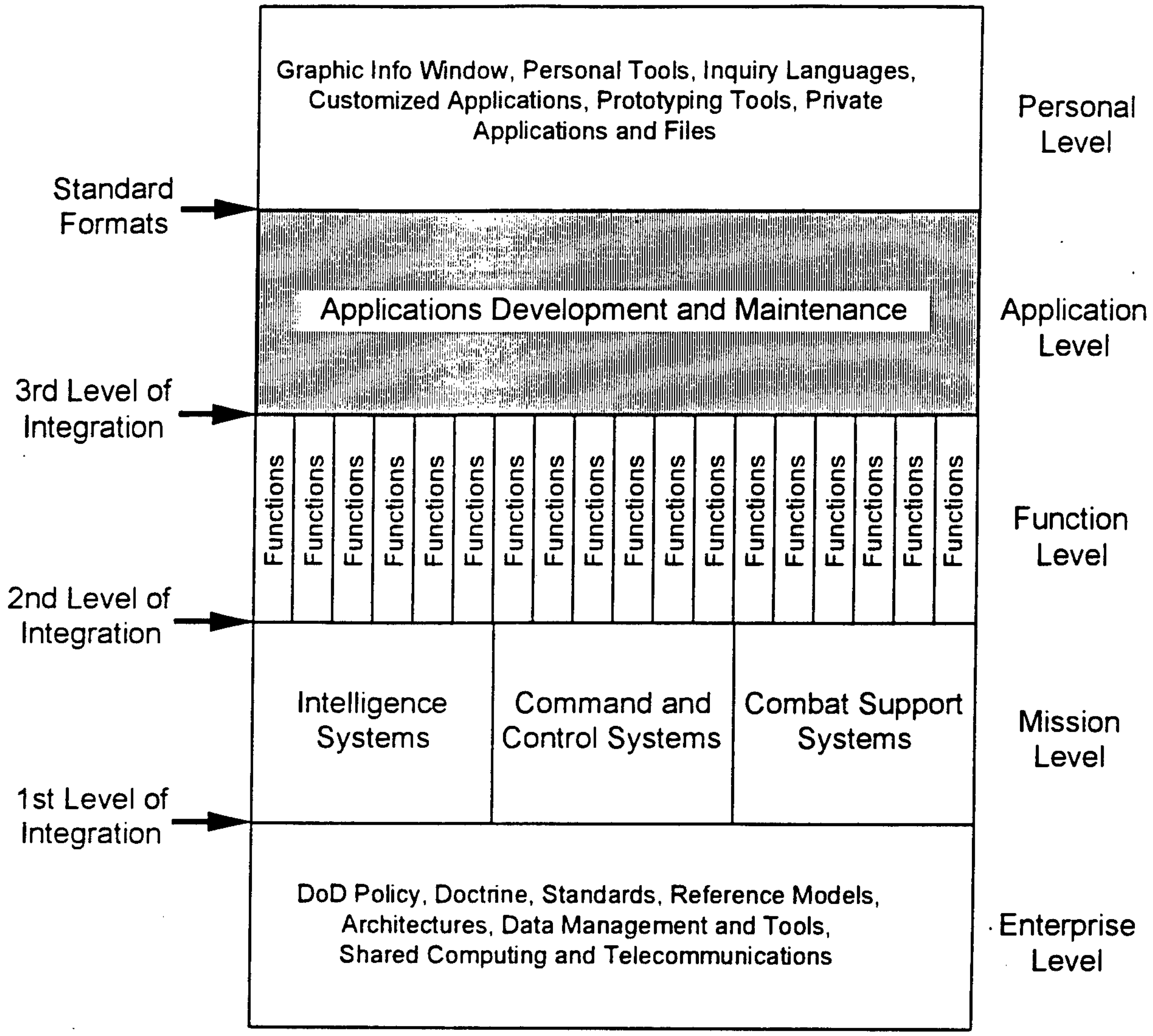

Graphic Info Window, Personal Tools, Inquiry Languages, Customized Applications, Prototyping Tools, Private Applications and Files
Personal Level
Standard Formats
Applications Development and Maintenance
Application Level
3rd Level of Integration
Functions
Function Level
2nd Level of Integration
Intelligence Systems
Command and Control Systems
Combat Support Systems
Mission Level
1st Level of Integration
DoD Policy, Doctrine, Standards, Reference Models, Architectures, Data Management and Tools, Shared Computing and Telecommunications
Enterprise Level

Figure 5-1. DoD IM Integration Model

- Functional integration: Functional integration generally involves collapsing two or more software modules that have similar functionality into a single new software module or involves relating two or more software modules with dissimilar functionality through a common database.

- Technical integration: Technical integration generally involves issues of compatibility and connectivity for interoperability of hardware and could involve software where relationships are involved (e. g., conversion between protocols).

5.4.1 The Enterprise Level

Level 1 is the Enterprise (or DoD-wide) Level. This level consists of integrating processes and procedures that are either manual or automated for all mission areas and their functions. Level 1 encompasses information management elements that are mandatory across the DoD. It includes IT and IM policy, procedures, standards, and doctrine that are established by the DoD or the Joint Chiefs of Staff (JCS). This level also includes standard IT capabilities such as technical and data standards, reference models and architectures, methods and tools, and shared computing and communications services. The integration and coordination of enterprise-level IT tasks support broad DoD policy and doctrine and are the responsibility of the Deputy Assistant Secretary of Defense (DASD) for IM. At this level, broad integration guidance and strategies for DoD information systems are established by the Defense Information Systems Agency (DISA) Joint Interoperability and Engineering Organization (JIEO).

The Enterprise Level is the foundation for standardizing technologies and services across the DoD. At this level, DISA develops common architectures, designs, and centrally manages the computer and communications utility. This utility is a global network that includes central processing resources, interoperable design activities, a DDRS and IDEF repository, shared databases, standards, central acquisition, security based on the DGSA, education and training, and other global and local common-use information technology services. The TAFIM is developed at this level to guide the development of the DoD technical architecture of this utility, to guide its use at other levels, and to promote total integration, interoperability, effectiveness, and efficiency including security of the DoD technical infrastructure through implementing DGSA concepts. When the TAFIM guidance and standards profile (and other DoD-wide architecture guidance such as the DGSA) are applied at other integration levels, DISA will review the resulting architecture products for conformance. The DGSA is a generic goal architecture that is designed as an integral part of the TAFIM guidance for the Enterprise Level.

5.4.2 The Mission Level

Level 2, the Mission Level, is composed of major DoD mission areas that are supported by systems for the mission areas such as Command and Control (C2) Systems, Intelligence Systems, and Combat Support Systems. (Combat support systems, formerly called business systems, include all systems that act as supporting elements for DoD.) At this level, areas of specialization and functional focus emerge, and mandatory DoD-wide technical requirements and capabilities are supplemented with mission-area specific requirements and capabilities. Strategy and planning for this level are developed under the direction of the DoD Principal Staff Assistants (PSA) and their appointed Functional Activity Program Managers (FAPMs) [DoD 8020.1-M].

At this level, DISA manages the integration of information systems functionality and technology within and across mission areas to achieve common major end-to-end functionality for command and control, intelligence, and business systems support. DISA tailors DoD-wide architectures, strategies, and plans for common use in networks, shared processing, and central design activities to satisfy mission-area requirements. For example, the TAFIM encourages tailoring to fit mission-area specific requirements of warfighters, intelligence analysts, and resource managers. JIEO prepares broad information system integration guidance for the development of information system integration strategies at the function level.

5.4.3 The Function Level

Level 3, the Function Level, includes multiple activities and processes of the DoD [DoD 8020.1-M]. At this level strategy and plans for these activities and processes are developed under the direction of PSAs or Principal Deputy Assistant Secretaries of Defense and their appointed FAPMs. Architectures are defined for the "to-be" functional operational practices and processes in accordance with DoD 8020.1-M and Change 1. Data models, activity models, and data architectures are developed to support simplified, streamlined, and improved practices and processes. Information system strategies and plans are developed that identify functional and technical requirements, priorities, schedules, and constraints for evolving information system baselines to the target information systems based on common architectures. In accordance with DoD IM policies and guidelines, DoD-wide and mission-area architectures are tailored to fit specific requirements, priorities, and constraints associated with unique functionality. The DoD Data Administrator (DA) and other elements of DISA work with the FAPM to ensure that functional data and information system strategies and plans conform to this guidance. They also review the Function Level architectures for conformance with DoD and mission-area architectures.

5.4.4 The Application Level

Level 4, the Application Level, includes the development, maintenance, and operation of information systems. In the integration concept each mission-area application can support a process, an activity, or a complete function. The application may execute on hardware bases that are distributed, shared, or dedicated. At this level, central design activities and data processing installations apply improved methods, tools, products, and services available through the activities of the Enterprise, Mission, and Function levels for design and development. Information systems are implemented by technical development activities in accordance with strategies and plans prepared at the function level.

5.4.5 The Personal Level

Level 5, the Personal Level, includes personal productivity tools and individual tailoring of automated capabilities for the end users. The tailoring must conform to guidelines and procedures that ensure the integrity of shared resources as well as effective operations in peacetime, transition to war, and war.

5.5 VIEWS OF THE INTEGRATION MODEL

The IM process is simultaneously a bottom-up and top-down process that is harmonized by new processes and procedures and technical integration support. As the cross-functional integration process takes hold, there will be a greater use of common architectures and "building blocks" managed at the enterprise and mission levels. Initially, however, process models, data models, standards, and information system architectures will be generated largely from a functional area and functional activity perspective to achieve immediate corporate IM objectives (e.g., migration toward system standardization). This reduces the need to develop new data, applications, and technical infrastructures. The two views are discussed below.

5.5.1 The Bottom-Up View

The bottom-up view is the foundation for each upper level of the integration model, which rests on a shared foundation of common policies, processes, procedures, methods, tools, and architectures. These elements are progressively tailored for specific mission areas, functionality, activities, and processes. Tailoring architectures promotes functional integration within and between the levels of the integration model. It helps ensure that users performing different functional activities work with systems that use a set of common architectures, standards, and services. Therefore, the users can use the planned global DoD network for meaningful information exchange and work together to achieve common objectives.

Figure 5-2 illustrates how the integration model can help achieve greater interoperability between functional activities in the DoD. The DoD is standardizing data and planning a global network at the Enterprise Level. The figure shows that different functional area applications will be able to access a common schema for shared databases maintained at the Enterprise Level and to use a global DoD network for information exchange. To the users of the functional activity applications, shared data will appear as part of the system they are using. Note, however, that each system may also have mission-area specific or unique data that may not be shared across functional lines.

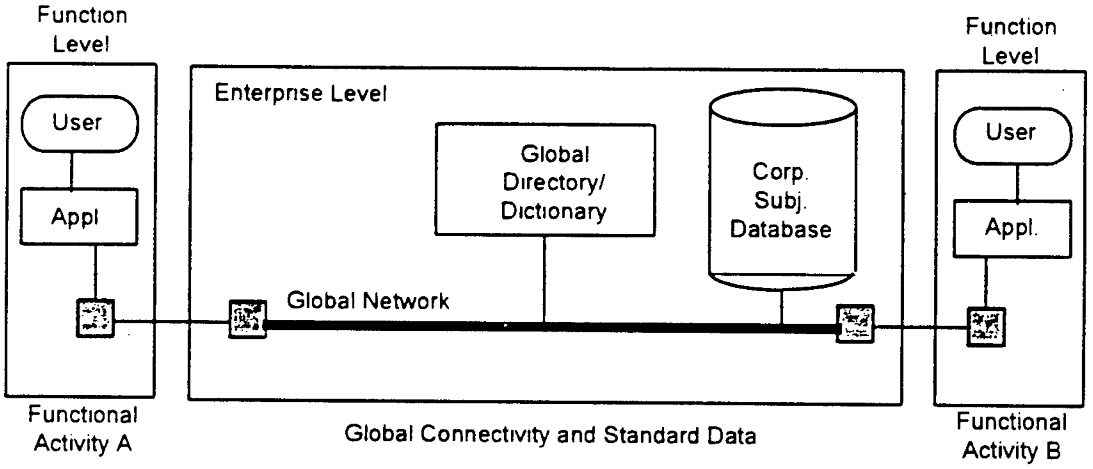


Figure 5-2. Example of Functional and Technical Integration

5.5.2 The Top-Down View

The top-down view of the integration model provides room for personal choice, innovation, and distributed development and control of systems by different organizations and individuals. The personal level can allow users to try out new ideas that may result in increased individual productivity. Procedures and technical controls will be used to control access to shared resources. The applications level develops, implements, and operates open systems using common methods, tools, and standards. Both shared and local applications can be developed. The function level provides the primary process models, data models, and information systems strategies for the DoD's functional activities. These elements are integrated into broader architectures that achieve cross-functional integration and interoperability. Each integration level inherits the characteristics of the upper integration levels.

5.6 ARCHITECTURE INTEGRATION AT LEVELS 1-3

Figure 5-3 shows the hierarchical structure of technical and other architectures, strategies, and plans that exist at each of the first three integration levels in the IM integration model. The architectures at lower levels guide and direct more specific architectures at the upper levels.

At Level 3, functional area activities can use a common architecture that is a subset of the functional area architecture. Functional areas can also use a common architecture that is a subset of the mission-area architecture.

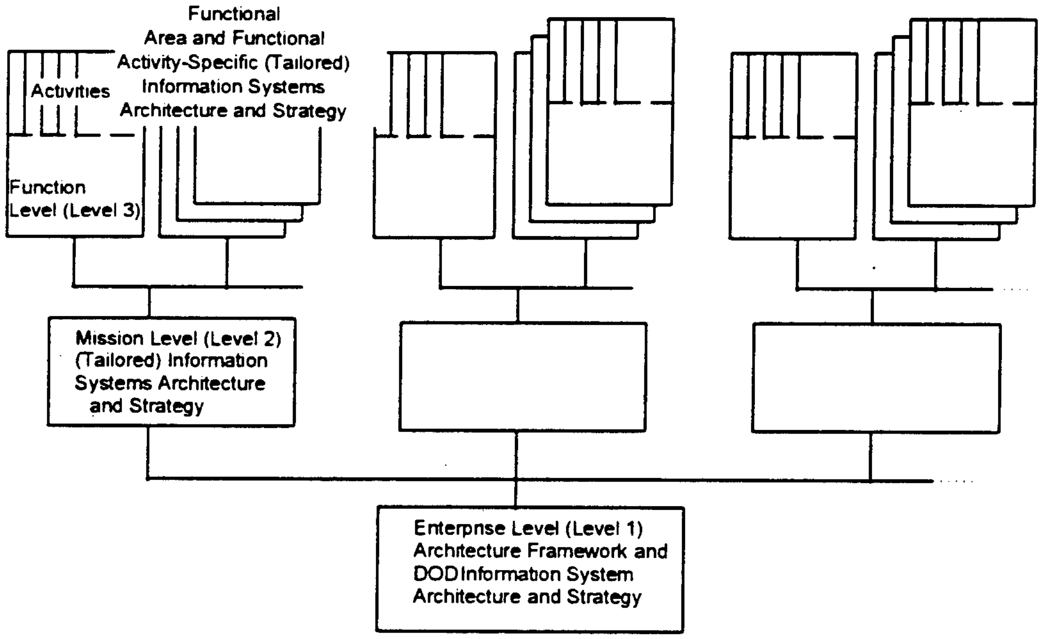

Functional Area and Functional Activity-Specific (Tailored) Information Systems Architecture and Strategy
Activities
Function Level (Level 3)
Mission Level (Level 2) (Tailored) Information Systems Architecture and Strategy
Enterprise Level (Level 1) Architecture Framework and DOD Information System Architecture and Strategy

Figure 5-3. Integration Levels of DoD IM Architectures and Strategies

At Level 2, mission areas, such as C2, can use a common architecture that is a subset of the overall DoD architecture.

At Level 1, the DoD Enterprise Level, a common information system architecture can be established that results in increased interoperability, integration, sharing of resources, and overall warfighting and support effectiveness.

The integration process for achieving interoperability is guided by the IM integration model, which consists of the following generic steps:

- Architectures, strategies, and technical management planning information are developed for each Functional Activity under the direction and guidance of the FAPM.

- Functional activity and functional area architectures, strategies, and technical management planning information are reviewed by the DA and DISA for conformance with enterprise (DoD-wide) and mission-area architectures, strategies, and technical management planning information.

- Interoperability requirements of the individual systems are translated to mission critical criteria for testing purposes. Interoperability testing verifies that mission critical criteria are met.

- Approved data, application software, infrastructure, and information system architectures, strategies, and technical management planning information become part of the overall enterprise and mission-area architecture baseline. They are subject to IM technical integration and configuration management policies and procedures. They form a basis for interoperability and operational testing as a precursor to system certification for interoperability.

- Cross-functional information system integration strategies and plans are developed at the enterprise and mission levels under the guidance and direction of the DASD (IM). DoD mission areas, vision, strategies, and plans will be translated into technical architectures, strategies, and plans to provide guidance for the functional level.

An iterative process involving the participation of PSAs at the Enterprise Level, the JCS, DISA, and the DoD Components aligns and reconciles the enterprise, mission areas, and functional level planning, architecture, and control processes.

Over time, the computer and communications utility will grow in scope and capability to provide an ever-increasing percentage of all information services for the DoD. In the long-term, functional users will obtain information services at affordable costs because of few new development requirements. Furthermore, once integration has been fully refined and institutionalized in a common infrastructure for DoD, system development efforts will speed up, and time between system conceptualization and operation will be greatly reduced.

This page intentionally left blank.

APPENDIX A

REFERENCES

Note: References appearing in this section represent documents used in preparation of the TAFIM, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

1. Executive Level Group (ELG) for Defense Information Management, 30 September 1990, A Plan for Corporate Information Management for the Department of Defense.

2. Department of Defense Directive (DoDD) 4630.5, 12 November 1992, Compatibility, Interoperability, and Integration of Command, Control, Communications, and Intelligence (C3I) Systems.

3. DoDD 8000.1, 27 October 1992, Defense Information Management (IM) Program.

4. DoDD 8120.1, 14 January 1993, Life-Cycle Management (LCM) of Automated Information Systems (AISs).

5. DoDD 8320.1, 26 September 1991, DoD Data Administration.

6. DoD Instruction (DoDI) 4630.8, 18 November 1992, Procedures for Compatibility, Interoperability, and Integration of Command, Control, Communications, and Intelligence (C3I) Systems.

7. DoDI 8120.2, 14 January 1993, Automated Information System (AIS) Life-Cycle Management (LCM) Process, Review, and Milestone Approval Procedures.

8. DoD 8020.1-M (Draft), August 1992 with Change 1 of January 1993, Functional Process Improvement (Functional Management Process for Implementing the Information Management Program of the Department of Defense) and Interim Management Guidance on Functional Process Improvement.

9. DoD 7920.2-M, March 1990, Automated Information System Life-Cycle Management Manual.

10. DoD CIM Functional Economic Analysis (FEA) Guidebook, (Draft), 15 January 1993.

This page intentionally left blank.

APPENDIX B

GLOSSARY

The glossary consists of two parts: Acronyms and Definitions.

ACRONYMS

AIS	Automated Information System
AITS	Adopted Information Technology Standards
AMWG	Architecture Methodology Working Group
API	Application Program Interface
APP	Application Portability Profile
ASC	Accredited Standards Committee
ASD(C3I)	Assistant Secretary of Defense for Command, Control, Communications, and Intelligence
ASIS	Ada Semantic Interface Specification
BBS	Bulletin Board System
C2	Command and Control
C3I	Command, Control, Communications, and Intelligence
CASE	Computer-Aided Software Engineering
CFA	Center for Architecture
CFII	Center for Integration & Interoperability
CIM	Corporate Information Management
CINC	Commander-in-Chief
CJCS	Chairman of the Joint Chiefs of Staff
CMP	Configuration Management Plan
COTS	Commercial-off-the-Shelf
DA	Data Administrator
DASD (IM)	Deputy Assistant Secretary of Defense for Information Management
DASP	Data Administration Strategic Plan
DDRS	Defense Data Repository System
DEPSECDEF	Deputy Secretary of Defense
DGSA	Department of Defense (DoD) Goal Security Architecture
DII	Defense Information Infrastructure

DISA	Defense Information Systems Agency
DISC	Defense Information System Council
DISN	Defense Information System Network
DISSP	Defense Information System Security Program
DoD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
DODM	DoD Manual
E-mail	Electronic Mail
EDI	Electronic Data Interchange
EEI	External Environment Interface
ELG	Executive Level Guidance
FAPM	Functional Activity Program Manager
FEA	Functional Economic Analysis
FIPS	Federal Information Processing Standard
HCI	Human Computer Interface
I-CASE	Integrated Computer-Aided Software Engineering
IEEE	Institute of Electrical and Electronic Engineers
IM	Information Management
IS	Information System
ISA	Information System Architecture
ISO	International Organization for Standardization
IT	Information Technology
ITRUS	Information Technology Reuse
ITSI	Information Technology Standards Information
JCS	Joint Chiefs of Staff
JIEO	Joint Interoperability and Engineering Organization
JTC	Joint Technical Committee
JTC3A	Joint Tactical Command, Control and Communications Agency
LAN	Local Area Network
LCM	Life-Cycle Management
MOE	Measures of Effectiveness
MS	Microsoft

N	Notarization
NATO	North Atlantic Treaty Organization
OASD	Office for the Assistant Secretary of Defense
OSD	Office of the Secretary of Defense
OSE	Open Systems Environment
OSI	Open Systems Interconnection
PMP	Program Management Plan
PSA	Principal Staff Assistant
STD	Standard
T&E	Test and Evaluation
TA	Technical Architecture
TAFIM	Technical Architecture Framework for Information Management
TBD	To Be Determined
TCP/IP	Transmission Control Protocol/Internet Protocol
TCSEC	Trusted Computer System Evaluation Criteria
TDI	Trusted Database Interpretation
TFA	Transparent File Access
TLSP	Transport Layer Security Protocol
TMP	Technical Management Plan
TNI	Trusted Network Interpretation
TP	Traffic Padding
TRM	Technical Reference Model
TRI-TAC	Tri-Service Tactical Communications Systems
TSIG	Trusted Systems Interoperability Group
U.S.	United States
WWW	World Wide Web

DEFINITIONS

Application–The use of capabilities (services and facilities) provided by an information system specific to the satisfaction of a set of user requirements. [P1003.0/D15]

Application Platform–The collection of hardware and software components that provide the services used by support and mission-specific software applications..

Application Portability Profile (APP)–The structure that integrates Federal, national, international, and other specifications to provide the functionality necessary to accommodate the broad range of Federal information technology requirements. [APP]

Application Program Interface (API)–(1) The interface, or set of functions, between the application software and the application platform. [APP] (2) The means by which an application designer enters and retrieves information.

Architecture–Architecture has various meanings depending upon its contextual usage. (1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE STD 610.12] (2) Organizational structure of a system or component. [IEEE STD 610.12]

Architecture: Baseline and Target–Defined and are significant parts of the technical management planning information (previously the technical management plan [TMP]). [DoD 8020.1-M with Change 1]

Architecture, Database–The logical view of the data models, data standards, and data structure. It includes a definition of the physical databases for the information system, their performance requirements, and their geographical distribution. [DoD 8020.1-M, Appendix J]

Architecture Target–Depicts the configuration of the target open information system. [DoD 8020.1-M]

Architecture, Infrastructure–Identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. [DoD 8020.1-M, Appendix J specifically paragraph 5(14)(c), Table J-2]

Architectural Structure–Provides the conceptual foundation of the basic architectural design concepts, the layers of the technical architecture, the services provided at each layer, the relationships between the layers, and the rules for how the layers are interconnected.

Automated Information System (AIS)–Computer hardware, computer software, telecommunications, information technology, personnel, and other resources that collect, record, process, store, communicate, retrieve, and display information. An AIS can include computer software only, computer hardware only, or a combination of the above. [DoDD 8000.1]

Availability–The probability that system functional capabilities are ready for use by a user at any time, where all time is considered, including operations, repair, administration, and logistic time. Availability is further defined by system category for both routine and priority operations. [JOPES ROC]

Baseline–A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and that can be changed only through formal change control procedures or a type of procedure such as configuration management. [IEEE STD 610.12]

Commercial-Off-the-Shelf (COTS)–Refers to an item of hardware or software that has been produced by a contractor and is available for general purchase. Such items are at the unit level or higher. Such items must have been sold and delivered to government or commercial customers must have passed customer's acceptance testing, be operating under customer's control, and within the user environment. Further, such items must have meaningful reliability, maintainability, and logistics historical data.

Communications Link–The cables, wires, or paths that the electrical, optical, or radio wave signals traverse. [TA]

Communications Network–A set of products, concepts, and services, that enable the connection of computer systems for the purpose of transmitting data and other forms (e.g., voice and video) between the systems.

Communications Node–A node that is either internal to the communications network (e.g., routers, bridges, or repeaters) or located between the end device and the communications network to operate as a gateway [TA]

Communications Services–A service of the Support Application entity of the Technical Reference Model (TRM) that provides the capability to compose, edit, send, receive, forward, and manage electronic and voice messages and real time information exchange services in support of interpersonal conferencing. [TA]

Communications System–A set of assets (transmission media, switching nodes, interfaces, and control devices), that will establish linkage between users and devices.

Configuration Management–A discipline applying technical and administrative direction and surveillance to: (a) identify and document the functional and physical characteristics of a configuration item, (b) control changes to those characteristics and, (c) record and report changes to processing and implementation status. [MIL-STD 973]

Connectivity Service–A service area of the External Environment entity of the Technical Reference Model that provides end-to-end connectivity for communications through three transport levels (global, regional, and local). It provides general and applications-specific services to platform end devices. [TA]

Database Utility Service–A Service of the Support Application Entity of the Technical Reference Model that provides the capability to retrieve, organize, and manipulate data extracted from a database. [TA]

Data Dictionary–A specialized type of database containing metadata, which is managed by a data dictionary system; a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases; an application of data dictionary systems. [DoDD 8320.1]

Data Element–A basic unit of information having a meaning and that may have subcategories (data items) of distinct units and values. [DoDD 8320.1]

Data Interchange Service–A service of the Platform entity of the Technical Reference Model that provides specialized support for the interchange of data between applications on the same or different platforms. [TA]

Data Management Service–A service of the Platform entity of the Technical Reference Model that provides support for the management, storage, access, and manipulation of data in a database. [TA]

Directory Service–A service of the External Environment entity of the Technical Reference Model that provides locator services that are restricted to finding the location of a service, location of data, or translation of a common name into a network specific address. It is analogous to telephone books and supports distributed directory implementations. [TA]

Distributed Database–(1) A database that is not stored in a central location but is dispersed over a network of interconnected computers. (2) A database under the overall control of a central database management system but whose storage devices are not all attached to the same processor. (3) A database that is physically located in two or more distinct locations. [FIPS PUB 11-3]

Enterprise–The highest level in an organization -- includes all missions and functions. [TA]

Enterprise Model–A high level model of an organization's mission, function, and information architecture. The model consists of a function model and a data model.

External Environment Interface (EEI)–The interface that supports information transfer between the application platform and the external environment. [APP]

Function–Appropriate or assigned duties, responsibilities, missions, tasks, powers, or duties of an individual, office, or organization. A functional area is generally the responsibility of a PSA (e.g., personnel) and can be composed of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviews). [Joint Pub 1-02, DoDD 8000.1, and DoD 8020-1M]

Functional Activity Program Manager (FAPM)–FAPMs are designated by PSAs and are accountable for executing the functional management process. Supported by functional representatives from the DoD Components, FAPMs develop functional architectures and strategic plans, and establish the process, data, and information system baselines to support functional activities within the functional area. [DoD 8020.1-M Ch 1 B(2)]

Functional Architecture–The framework for developing applications and defining their interrelationships in support of an organization's information architecture. It identifies the major functions or processes an organization performs and their operational interrelationships. [DoD 5000.11-M]

Functional Area–A range of subject matter grouped under a single heading because of its similarity in use or genesis. [DoDD 8320.1]

Functional Data Administrator (FDAd)–Office of the Secretary of Defense (OSD) PSAs exercise or, designate functional data administrators to perform data administrator responsibilities to support execution of the functional management process, and to function within the scope of their overall assigned responsibilities. [DoDD 8320.1 and DoD 8020.1-M, Appendix A].

Functional Economic Analysis (FEA)–A structured proposal that serves as the principal part of a decision package for enterprise (individual, office, organization -see function) leadership. It includes an analysis of functional process needs or problems; proposed solutions, assumptions, and constraints; alternatives; life-cycle costs; benefits and/or cost analysis; and investment risk analysis. It is consistent with, and amplifies, existing DoD economic analysis policy. [DoDI 7041.3, DoDD 8000.1, and DoD 8020.1-M, Appendix H]

Hardware–(1) Physical equipment, as opposed to programs, procedures, rules, and associated documentation. (2) Contrast with software [FIPS PUB 11-3]

Information–Any communication or representation of knowledge such as facts, data, or opinions, in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual forms. [OMB CIRC A-130]

Information Domain–A set of commonly and unambiguously labeled information objects with a common security policy that defines the protections to be afforded the objects by authorized users and information management systems. [DISSP]

Information Management (IM)–The creation, use, sharing, and disposition of information as a resource critical to the effective and efficient operation of functional activities. The structuring of functional processes to produce and control the use of data and information within functional activities, information systems, and computing and communications infrastructures. [DoDD 8000.1]

Information Resources Management (IRM)–The planning, budgeting, organizing, directing, training, promoting, controlling, and management activities associated with the burden (cost), collection, creation, use, and dissemination of information by Agencies and includes the management of information and related resources, such as Federal information processing (FIP) resources. [PL No 99-591, DoDD 8000.1.]

Information Technology (IT)–The technology included in hardware and software used for Government information, regardless of the technology involved, whether computers, communications, micro graphics, or others. [OMB Circular A-130 and DoDD 8000.1.]

Infrastructure–Infrastructure is used with different contextual meanings. Infrastructure most generally relates to and has a hardware orientation but note that it is frequently more comprehensive and includes software and communications. Collectively, the structure must meet the performance requirements of and capacity for data and application requirements. Again note that just citing standards for designing an architecture or infrastructure does not include functional and mission area requirements for performance. Performance requirement metrics must be an inherent part of an overall infrastructure to provide performance interoperability and compatibility. It identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. [DoD 8020.1-M]

Integration–Integration is the result of an effort that joins two or more similar products such as individual system elements, components, modules, processes, databases, or other entities, and produces a new product that functions, as a replacement for the two or more similar but less capable entities (products), in a framework or architecture in a seamless manner. Institute of Electrical and Electronic Engineers (IEEE) Standard (STD) 610.12 defines an "integration architecture" as a framework for combining software components, hardware components, or both into an overall system. [IEEE STD 610.12]

Interoperability–(1) The ability of two or more systems or components to exchange and use information. [IEEE STD 610.12]. (2) The ability of the systems, units, or forces to provide and receive services from other systems, units, or forces, and to use the services so interchanged to enable them to operate effectively together. The conditions achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. [Joint Pub 1-02, DoD/NATO] [JOPES ROC]

Legacy Environments–Legacy environments could be called legacy architectures or infrastructures and as a minimum consist of a hardware platform and an operating system. Legacy environments are identified for phase-out, upgrade, or replacement. All data and applications software that operate in a legacy environment must be categorized for phase-out, upgrade, or replacement.

Legacy Systems–Systems that are candidates for phase-out, upgrade, or replacement. Generally legacy systems are in this category because they do not comply with data standards or other standards. Legacy system workloads must be converted, transitioned, or phased out (eliminated). Such systems may or may not operate in a legacy environment.

Life Cycle–The period of time that begins when a system is conceived and ends when the system is no longer available for use. [IEEE STD 610.12] AIS life cycle is defined within the context of life-cycle management in various DoD publications. It generally refers to the usable system life.

Local Area Network (LAN)–A data network, located on a user's premises, within a limited geographic region. Communication within a local area network is not subject to external regulation; however, communication across the network boundary may be subject to some form of regulation. [FIPS PUB 11-3]

Migration Systems–An existing AIS, or a planned and approved AIS, that has been officially designated to support common processes for a functional activity applicable to use DoD-wide or DoD Component-wide. Systems in this category, even though fully deployed and operational, have been determined to accommodate a continuing and foreseeable future requirement and, consequently, have been identified for transitioning to a new environment or infrastructure. A migration system may need to undergo transition to the standard technical environment and standard data definitions being established through the Defense IM Program, and must "migrate" toward that standard. In that process it must become compliant with the Reference Model and the Standards Profile. A system in this category may require detailed analysis that involves a total redesign, reprogramming, testing, and implementation because of a new environment and how the "users" have changed their work methods and processes. The detailed analysis may identify the difference between the "as is" and the "to be" system. [DoD 8020.1-M.]

Multimedia Service–A service of the TRM that provides the capability to manipulate and manage information products consisting of text, graphics, images, video, and audio. [TA]

Open Specifications–Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards. [P1003.0/D15]

Open System–A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [P1003.0/D15]

Open Systems Environment (OSE)–The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of applications, data, or people, as specified by information technology standards and profiles. [P1003.0/D15]

Operating System Service–A core service of the Platform entity of the Technical Reference Model that is needed to operate and administer the application platform and provide an interface between the application software and the platform (e.g., file management, input/output, print spoolers). [TA]

Platform–The entity of the Technical Reference Model that provides common processing and communication services that are provided by a combination of hardware and software and are required by users, mission area applications, and support applications. [TA]

Portability–(1) The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE STD 610.12] (2) A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment. [IEEE TUTOR] (3) The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TA]

Process Model–Provides a framework for identifying, defining, and organizing the functional strategies, functional rules, and processes needed to manage and support the way an organization does or wants to do business -- provides a graphical and textual framework for organizing the data and processes into manageable groups to facilitate their shared use and control throughout the organization. [DoD 5000.11-M]

Profile–A set of one or more base standards, and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [P1003.0/D15]

Profiling–Selecting standards for a particular application. [P1003.0/D15]

Response Time–The ability to react to requests within established time criteria. To be operationally effective, the system must product the desired output in a timely manner based on system category for routine or priority operations. [JOPES ROC]

Scalability–The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII] The capability to grow to accommodate increased work loads.

Seamless Interface–Ability of facilities to call one another or exchange data with one another in a direct manner. Integration of the user interface that allows a user to access one facility through another without any noticeable change in user interface conventions. [DSAC SYS IM]

Stovepipe System–A system, often dedicated or proprietary, that operates independently of other systems. The stovepipe system often has unique, nonstandard characteristics.

System–People, machines, and methods organized to accomplish a set of specific functions. [FIPS PUB 11-3]

System Management Service–A service of the Platform entity of the TRM that provides for the administration of the overall information system. These services include the management of information, processors, networks, configurations, accounting, and performance. [TA]

Technical Reference Model (TRM)–The document that identifies a target framework and profile of standards for the DoD computing and communications infrastructure. [TRM]

User–(1) Any person, organization, or functional unit that uses the services of an information processing system. (2) In a conceptual schema language, any person or any thing that may issue or receive commands and messages to or from the information system. [FIPS PUB 11-3]

User Interface Service–A service of the Platform entity of the Technical Reference Model that supports direct human-machine interaction by controlling the environment in which users interact with applications. [TA]

This page intentionally left blank.

APPENDIX C

VISION FOR DOD INFORMATION MANAGEMENT

Section 3.0 focused on the vision for information technology. This appendix focuses on the vision for information management. A significant aspect of Section 3.0 addresses the management of information technology. Overall, the visions include the use of information technology to manage information. For example, information technology enables functional managers to standardize and streamline processes and activities, reduce non-value-added work, improve productivity, and lower costs for operations across the DoD. Information management is critical to providing efficient and effective information functional processes and practices across the DoD. It is recognized as a force effectiveness and support multiplier during peacetime preparedness, transition to war, and war. The integration of information management principles with technologies into all aspects of DoD operations means that effective military capability is maintained while defense budgets decline.

Functional methods and measures are being updated and documented across the DoD. Options and opportunities to standardize, simplify, and improve processes and management practices will be identified and selected at all levels using process modeling, process improvement, and functional economic analysis methods.

Measures of performance will be used to manage functions and systems resulting in improved quality, productivity, cost performance, and functionality. The mechanisms to capture performance data are built into information systems, enabling managers to evaluate their effectiveness and make continuous improvements. Comprehensive evaluations will be performed continuously throughout the system life cycle to ensure the systems continue to meet the functional needs of the users

Data standards are being established and implemented across the mission areas. A data modeling initiative will result in providing standard data descriptions and attributes captured in a DoD-wide Defense Data Repository System (DDRS). With common data definitions, data reuse will become the standard practice in all systems development and maintenance. All forms of data, including alphanumeric, geographic, document format, and multi-media are managed for interoperability and meaningful exchange within and across functions. Standard data definitions and models are being developed with industry and other parts of the Federal Government.

DoD will implement shared corporate databases that capture, store, and maintain standard data. Data will be input at the source for accuracy and validity and reused whenever possible. Horizontal and perpendicular data transformations will be controlled and included in the data repository. Data will be input through a variety of flexible and responsive devices and mechanisms from the office to the battlefield. Electronic capture and display of information, which is becoming normal practice, will lead to a "less-paper" (and in some cases a "paper-less")

DoD environment. Currency, reliability, and responsiveness are being greatly improved, errors avoided, and the integrity and security of DoD data will be assured by new procedures and automation.

Users will eventually access data through a common global network, and through other media such as CD-ROM, limited only by their need to know. The physical location of data will become transparent to users and applications. A DoD directory and dictionary capability maintains global and functional schemas for the corporate database. A total information management facility will be established to filter, process, distribute, and fuse information when and where it is needed.

Electronic data interchange (EDI) of all forms of information is planned and will be implemented following the world-wide lead of industry. Transaction systems that automatically process specific tasks will be common. These capabilities will reduce manual work, eliminate errors, and improve the performance of complex operational activities. For example, DoD will routinely conduct most of its business with industry suppliers through electronic commerce and technical document interchange. Artificial intelligence will become critical to many functions, enabling processes to be substantially automated.

The foundation of standard processes and data, and new technologies, will enable a variety of typical functions to be performed far more effectively and efficiently. For example,

- Office automation will benefit from a suite of standards-based, flexible and integrated word processing, graphics, document preparation, and groupware applications.

- Decision support to managers and commanders will provide benefits from video-conferencing (to the desktop when necessary), mail services, briefing preparation and display facilities, and modeling and simulation capabilities.

- The operational commander will benefit from the DoD-wide technical capabilities to pull, fuse, filter, and disseminate the precise information needed to address situation-dependent missions.

- A rapid, responsive, efficient, and quality-oriented AIS life-cycle development and maintenance process is being instituted. This process is based on certain key practices such as:

  - Process modeling and functional economic analysis

  - Data administration procedures, practices, and standard data elements in a DoD DDRS

  - Open systems environments, architectures and implementations

  - Integrated computer-aided software engineering (CASE) methods and tools

- Streamlined software processes, metrics, and reuse
- Streamlined information technology reuse and acquisition
- Shared design, processing, network, and information center
- Services (i.e., a utility) delivered on a fee-for-service basis.

The roles and responsibilities of functional and technical managers, developers, and operators have been structured to leverage the strengths of each. Technical integration management support to functional activity managers is key to helping them plan integrated information systems support within and across functions. Information technologists provide the required tools and building blocks needed to develop, install, and operate efficient and effective information systems.

This page intentionally left blank.

APPENDIX D

DOD MEMORANDA ADDRESSING USE OF THE TAFIM

This appendix contains the text of three DoD memoranda that address the use of the TAFIM:

- 30 March 1995 Memorandum from the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence
- 12 November 1993 Memorandum from the Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (with attachment)
- 13 October 1993 Memorandum from the Deputy Secretary of Defense (with attachment).

This page intentionally left blank.

MEMORANDUM FROM THE ASSISTANT SECRETARY OF DEFENSE

March 30, 1995

MEMORANDUM FOR UNDER SECRETARIES OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY (RD&A)
ASSISTANT SECRETARY OF THE NAVY (RD&A)
ASSISTANT SECRETARY OF THE AIR FORCE
(ACQUISITION) (SAF/AQ)
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM), Version 2.0

My memorandum dated June 23, 1994 established the TAFIM as the single framework to promote the integration of Department of Defense (DoD) information systems, expanding the opportunities for interoperability and enhancing our capability to manage information resources across the Department. The latest version of the TAFIM, Version 2.0, is complete and fully coordinated. Version 2.0 consists of seven volumes as shown in the attachment. The TAFIM will continue to guide and enhance the evolution of the Department's information systems technical architectures.

I want to reiterate two important points that I made in my June 1994 memorandum. First, the Department remains committed to a long range goal of an open systems environment where interoperability and cross functional integration of our systems and portability/reusability of our software are key benefits. Second, the further selection and evaluation of migration systems should take into account this long range goal by striving for conformance to the TAFIM to the extent possible.

Effectively immediately, new DoD information systems development and modernization programs will conform to the TAFIM. Evolutionary changes to migration systems will be governed by conformance to the TAFIM.

The TAFIM is maintained by the Defense Information Systems Agency (DISA) and is available electronically via the DISA On-Line Standards Library. Hardcopy is available through the Defense Technical Information Center. The TAFIM is an evolving set of documents and comments for improving may be provided to DISA at any time. The DISA action officer is Mr. Bobby Zoll, (703) 735-3552. The OSD action officer is Mr. Terry Hagle, (703) 604-1486.

s/Emmett Paige, Jr.

MEMORANDUM FROM
THE ASSISTANT SECRETARY OF DEFENSE

November 12, 1993

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
DIRECTOR, DEFENSE RESEARCH AND ENGINEERING
ASSISTANT SECRETARIES OF DEFENSE
COMPTROLLER
GENERAL COUNSEL
INSPECTOR GENERAL
DIRECTOR, OPERATIONAL TEST AND EVALUATION
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTOR OF ADMINISTRATION AND MANAGEMENT
DIRECTORS OF THE DEFENSE AGENCIES

SUBJECT: Selection of Migration Systems

This memorandum provides the generic evaluation criteria to be used in selection of migration systems as required by the Deputy Secretary of Defense (DEPSECDEF) memorandum of 13 October 1993, "Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement." The Department of Defense (DoD) must improve the quality and effectiveness of information support for our fighting forces, reduce the cost of duplicative processes, eliminate nonessential legacy systems in all functional areas, and minimize the cost and difficulty of information systems technical integration. Information systems are comprised of applications, data and infrastructure. Expedited selection of migration systems has been established by the Deputy Secretary of Defense as a matter of urgency throughout the DoD. Selection shall be based on these four factors:

- Functional: To be selected as a migration system, the information system will have to be based on defined work processes and will have to be based on the degree to which the system meets the information needs of users within and across functional areas. A decision should be generally supported by the functional user community within the DoD Components, including the Chairman of the Joint Chiefs of Staff (CJCS) representing the unified combatant commands.

- Technical: The system can evolve (migrate) to be supported by the integrated, standards-based architecture prescribed for the future Defense Information Infrastructure (DII).

- Programmatic: A functional economic analysis that documents a reasonable range of alternatives that meet both functional and technical objectives is required. The alternatives must be within programmatic constraints (resources, schedules, and acquisition strategy), and justify adopting the migration system to the Department. Given the compressed time frames, the PSAs may elect to base their migration decision on an abbreviated functional economic analysis. Acquisition strategy planning factors will be considered in accordance with Acting ASD(C^3I) memorandum of February 4, 1993, "Acquisition Strategy Planning for CIM Migration Systems."

- Data: The ability to transition to data standards is a fundamental requirement for an information system in order for it to be selected as a migration system. Applications should lend themselves to data sharing within their design. Migration plans must include transition to DoD standard data and shared data concepts.

Migration systems selection procedures and factors are discussed in our Interim Management Guidance on Functional Process Improvement (August 5, 1992, and January 15, 1993). Except where exempted under DoD Directive 8120.1, Section B, the selection procedures apply to all AISs in the Department. This includes all C^3I systems except those specifically and individually exempted by me in accordance with my DoD Senior Information Management (IM) authority under DoD Directives 5137.1 and 8000.1. All information technology services shall be transition to the selected migration systems over a period not to exceed three years, and the legacy systems providing these services shall be terminated. Any funding for development, modernization, or enhancement of these legacy systems requires the approval of the DoD Senior IM Official, in accordance with the DEPSECDEF's memorandum of October 13, 1993. Life-cycle management reviews of migration systems shall also address these candidate legacy systems and data until their termination.

Migration system selection shall be made by the Office of the Secretary of Defense (OSD) Principal Staff Assistant(s) (PSAs), or CJCS, having functional responsibility for the missions and functions supported by the system, with the participation of affected DoD Components. The choice of functional criteria guidance in the selection of migration systems is the responsibility of the PSAs/CJCS. As the DoD Senior IM Official, I shall approve the proposed selection, based on my review of the selecting official's evaluation of technical, programmatic, and data factors. Because technical factors are critical to successful implementation of the DII, I shall have additional studies conducted where appropriate, and I shall withhold my approval where significant issues remain unresolved. Disagreements shall be resolved in accordance with DoD Directive 8000.1, Section E.1.d.

Attached to this memorandum are key technical considerations that must be addressed in the selection process. Assistance in your selection of migration systems and in preparation of the appropriate documentation is available through the Defense Information Systems Agency Center for Integration and Interoperability. If you would like this assistance, please contact Dr. Michael Mestrovich at (703) 756-4740.

s/Emmett Paige, Jr.

Attachment

KEY TECHNICAL FACTORS TO BE CONSIDERED IN THE SELECTION OF MIGRATION SYSTEMS

Technical Factors

Extent to which the candidate legacy automated information system (including Command, Control, Communications and Intelligence (C^3I) systems) currently conforms to, or can evolve (migrate) to conformance with, the open systems environment and standards-based architecture defined by the DoD Technical Architecture Framework for Information Management (TAFIM)[1].

Difficulty, cost, and time line for migrating the system (including its applications, data, and supporting infrastructure) as expeditiously as possible from its current technical environment to conformance with:

- The TAFIM.
- DoD standard data, based on the DoD Data Model. The DoD Data Model is a principal component of the DoD Enterprise Model.
- Shared use of applications, databases, and the computing and communications infrastructure with other designated migration systems.
- Cost effective, timely, secure, and highly reliable support to all functional users from consolidated data processing facilities.

Timeliness, completeness, and availability of life-cycle management and supporting documentation, particularly including data and application software documentation.

Difficulty, cost, and time line for application of:

- DoD information technology utility services.
- Commercial-off-the-shelf (COTS) software, and portable, re-usable software modules.
- Ada and computer-aided software engineering (CASE) tools and methods.

Current and future interface, interoperability, and integration requirements with other systems and databases within and across all DoD functional activities and functional areas.

[1] Office of the Assistant Secretary of Defense (C^3I) Memorandum, "Interim Management Guidance on the Technical Architecture Framework for Information Management (TAFIM)," January 15, 1993.

Application of Technical Factors

Application of these technical factors results in giving preference to systems that:

- Have been developed using Ada and other "state of the industry" software engineering best practices, are well documented, and are under good configuration control.

- Use current COTS information technology software and hardware, such as data dictionaries and data base management systems, optical disk technology, etc.

- On the whole, are more compliant rather than less compliant with the technical factors listed above, and apply those factors consistently across all systems supporting the functional area.

Assessment and Plans

The selection of a candidate migration AIS must be founded on its functional and technical adequacy. Migration assessment includes a technical analysis of migration candidate systems to ensure legacy applications will meet the information requirements of the functional user and that has the ability to accommodate subsequent functional and technical improvement activities.

A migration plan consisting of functional, technical and data concerns, with programmatic considerations is the start of the process for selecting migration systems. The DoD "Tree" diagrams, a quarterly publication from DISA/Center for Integration and Interoperability (CFII), displays each functional area's decisions for integrating. These "Tree" diagrams will be completed by all functional areas with target dates to depict the Enterprise Integration. The diagrams present an important migration picture but stop short of the migration planning that is necessary for implementation. The DISA/CFII is available to help each functional area develop migration plans and assess technical cross-functional integration for the Enterprise.

To validate the technical sufficiency of a candidate migration system, the applications should be evaluated in terms of relevant functional, technical, data handling, and programmatic criteria.

ATTACHMENTMEMORANDUM FROM THE DEPUTY SECRETARY OF DEFENSE

13 October 1993

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
ASSISTANT TO SECRETARIES OF DEFENSE
COMPTROLLER
GENERAL COUNSEL
INSPECTOR GENERAL
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTOR OF ADMINISTRATION AND MANAGEMENT
DIRECTORS OF THE DEFENSE AGENCIES

SUBJECT: Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement

My May 7, 1993, memorandum reiterated the full commitment of the Department of Defense (DoD) to the "... improvements, efficiencies, and productivity that are the essence of CIM." The focus of Corporate Information Management (CIM) on functional process improvement, migration systems, and data standardization has my full support. We need to get on with the job. In order to offset our declining resources, we must accelerate the pace at which we define standard baseline process and data requirements, select and deploy migration systems, implement data standardization, and conduct functional process improvement reviews and assessments (business process re-engineering) within and across all functions of the Department. The acceleration of these actions is key to containing the functional costs of performing the DoD mission within our constrained budget.

The attached guidance requires that addressees expedite selection of standard migration systems and standard data as the basis for process improvement reviews and assessments. The attached guidance expands on direction previously issued by the Comptroller on June 25, 1990, and by the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence) (ASD(C^3I)) on February 11, 1991. The ASD(C^3I) will work with you to ensure that overall functional and Component requirements are met and balanced as we integrate and improve systems, data, and processes across the DoD. Our near-term strategy requires:

- Selection of migration systems within six months, with follow-on DoD-wide transition to the selected systems over a period not to exceed three years.

- Complete data standardization within three years by simplifying data standardization procedures, reverse engineering data requirements in approved and proposed migration systems, and adopting standard data previously established by individual functions and Components for DoD-wide use wherever practical.

The above actions should be implemented immediately, and given appropriate priority in your current and future resource planning and allocation.

Ongoing information management initiatives such as functional process improvement projects, functional and technical integration analysis and planning, and software engineering methods modernization should continue on an expedited basis. However, completion of these current initiatives will not be prerequisites to implementation of the migration system and data standards acceleration strategy. Once standard DoD-wide process, system, and data baselines are established, process improvement studies will be more productive and study results can be more rapidly implemented.

It is understood that the implementation of standard migration systems may result in the loss of automated functionality by selected system users, whereas others may gain functionality. Loss of functionality should not be used as a reason to delay migration system selection and deployment unless there is a documented adverse impact on readiness within the deployment period, or an inability to comply with the law.

The ASD(C^3I) is responsible for supplementing existing procedures with generic evaluation criteria within 30 days to be used in selecting migration systems, and ensuring the objectivity of the selection process.

I request that you personally ensure these actions are accomplished on schedule, and that you report to me on your progress by January 31, 1994.

s/William J. Perry

Attachment

DEPARTMENT OF DEFENSE

STRATEGY FOR ACCELERATION OF MIGRATION SYSTEMS AND DATA STANDARDS

OBJECTIVE

Improve the quality and utility of DoD information while reducing the annual cost of DoD operations.

STRATEGY

Migration Systems

- OSD Principal Staff Assistants, together with their Defense Component counterparts, will, by March 31, 1994, select an information system(s) for each of their respective functional areas of responsibility for designation as the standard, DoD-wide migration system.

- Concurrently, OSD Principal Staff Assistants will develop plans to transition all information technology services throughout the DoD to the selected migration systems, over a period not to exceed three years. Draft plans will be circulated to other Principal Staff Assistants and to Defense Components so that cross-functional and other implementation issues can be identified for consideration by functional and Defense Component members of the DoD corporate Functional Integration Board, chaired by the Deputy Assistant Secretary of Defense (Information Management).

- Funding for development, modernization, or enhancement of legacy systems not selected to be migration systems will be stopped except where approved by the DoD Senior Information Management Official as absolutely essential to support DoD missions or comply with the law.

- The plan for implementing and transitioning services to the selected migration systems should simultaneously forecast a schedule, to the extent practical, for incorporating within the migration systems:

  - Improved functionality and cross-functional integration based on accelerated process improvement reviews and assessments.

  - Interoperability, technical integration, DoD standard data, and integrated databases to provide higher quality and lower cost information technology services for all users.

- Where a requirement is demonstrated to develop a follow-on, new start system to replace the standard migration system in order to meet CIM objectives and the information management policies and principles established in DoD Directive 8000.1, OSD Principal

Staff Assistants will conduct the necessary process improvement studies to develop functional requirements within the next three years.

Data Standardization

- Each DoD Principal Staff Assistant, together with their Defense Component counterparts, will develop and execute a plan in accordance with DoD Directive 8320.1 to standardize the data elements for which they are the custodian within the next three years.

- The ASD(C^3I) will, by January 31, 1994, develop simplified and streamlined processes for data standardization and data administration within the DoD.

- In the interim, the Department will continue to use the existing standard data elements within each function and Defense Component that have been developed under previous procedures. These interim standard data elements are the data standards until replaced by those prepared under DoD Directive 8320.1.

DEFINITIONS

The definitions below are intended to clarify the terms used in the DoD near-term strategy for acceleration of migration systems and data standards. Formal definitions are published in DoD directives or other publications.

Baseline Processes and Data

A baseline is something that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. Baseline processes and data establish how a function operates today (the "as is" environment), and what current functional requirements must be satisfied by the supporting migration system. Process improvement projects assess the "as is" baseline to determine what improvements should be made (to the "to be" environment). Once these improvements have been implemented, they define a new process and data baseline for the next iteration of improvements.

Data Standard (also called standard data)

A data element that has been through a formal analysis (called "data standardization") to reach agreement on its name, meaning, and characteristics, as well as its relationship to other standard data elements. Much like a common language, data standards enable processes and their supporting information systems to be integrated across functions, as well as within them, and improve the quality as well as the productivity of enterprise performance.

Data Standardization

The process of reviewing and documenting the names, meanings, and characteristics of data elements so that all users of the data have a common, shared understanding of it.

Data standardization is a critical part of the DoD Data Administration Program, managed under DoD Directive 8320.1. Data administration is the function that manages the definition and organization of the Department's data.

Function

Appropriate or assigned duties, responsibilities, and tasks that produce products or provide services. In the DoD, a functional area (e.g., personnel) is comprised of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviewing candidates). The functions of the DoD are the responsibility of designated officials who exercise authority over organizations set up to accomplish their assigned functions. The structure and interrelationships among DoD functions and standard data are documented in the DoD Enterprise Model.

Individual functions within the DoD rely on other functions for products and services. In a large, complex enterprise such as the Department of Defense, functions must work together to support the mission of the enterprise; this significantly increases the importance of cross-functional programs, such as data standardization.

Functional Process Improvement (also called business process re-engineering)

Application of a structured methodology to define a function's objectives and a strategy for achieving those objectives; its "as is" and "to be" process and data environments; its current and future mission needs and end user requirements; and a program of incremental and evolutionary improvements to processes, data, and supporting migration systems that are implemented through functional, technical, and economic analysis and decision-making.

Procedures for conducting process improvement reviews and assessments in the DoD are provided in OASD(C^3I) memoranda on Interim Management Guidance on Functional Process Improvement (August 5, 1992, and January 15, 1993).

Integration

Explicit top management initiatives to ensure that interdependent functions or systems operate effectively and efficiently for the overall benefit of the enterprise (i.e., the DoD). This contrasts with coordination among functions or systems, which ensures non-interference, but does not provide integration.

"Integration" implies seamless, transparent operation based on a shared or commonly-derived architecture (functional or technical) and standard data. "Interoperability" implies only the ability of a function or system to exchange information or services with another, separate function or system using translators or interchange rules/standards.

Migration System

An existing automated information system (AIS), or a planned and approved AIS, that has been officially designated as the single AIS to support standard processes for a function. Other AISs, called "legacy systems," that duplicate the support services provided by the migration system are terminated, so that all future AIS development and modernization can be applied to the migration system. A migration system is designated (or selected) by the OSD Principal Staff Assistant(s) and their Defense Component counterparts whose function(s) the system supports, with the coordination of the DoD Senior Information Management Official.

Upon selection and deployment, the migration system becomes the single AIS baseline for:

- Incremental and evolutionary changes that are required to implement functional process improvements, or to execute additional responsibilities assigned to the function that the system supports.

- Technical enhancements that implement standard data and integrated databases, and that migrate the system toward an open systems environment and a standards-based architecture defined by the DoD Technical Architecture Framework for Information Management.

Requirements for selection of migration systems are identified in Chapters 6 and 7 of OASD(C^3I) memoranda on Interim Management Guidance for Functional Process Improvement (August 5, 1992, and January 15, 1993); these procedures should be tailored as appropriate to facilitate expeditious selection. Subsequent development and modernization of migration systems is accomplished in accordance with DoD Directive 8120.1 and DoD Instruction 8120.2.

This page intentionally left blank

APPENDIX E

PROPOSING CHANGES TO TAFIM VOLUMES

E.1 INTRODUCTION

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the Configuration Management Plan (CMP), and the Program Management Plan (PMP)). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section E.2. Guidance on the use of the format is provided in Section E.3.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via electronic mail (E-mail) in American Standards Code for Information Interchange (ASCII) format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown in Section E.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail: **TAFIM**
Booz•Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/824-3770**, indicate "TAFIM" on cover sheet.

E.2 TAFIM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

E.3 FORMAT GUIDANCE

The format in Section E.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of**

Contact Identification part **(E.2 a)** of the format would identify that individual. The information in the **Document Identification** part of the format **(E.2 b)** is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts **(E.2 c, E.2 d, or E.2 n)** of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: 'The working baseline will only be available to the TAFIM project staff.'" The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The E.2 c (5), E.2 d (5), or E.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in E.2 c (6), E.2 d (6), or E.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

This page intentionally left blank.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 2: Technical Reference Model

Version 3.0

30 April 1996

FOREWORD: ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

- This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents

- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.

- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 2 of the TAFIM, contains significant revisions from the Version 2.0 edition of this volume. These changes are the result of a harmonization of Volumes 2, 4, and 7 conducted by MITRE for the Defense Information Systems Agency (DISA) Joint Interoperability and Engineering Organization (JIEO) Center for Standards (CFS). All of the proposed harmonization changes could not be implemented at this time because of the time and resources available and the level of consensus within the Architecture Methodology Working Group (AMWG). Volume 2 has been updated with those changes that could be accomplished within the time available. The major changes to Volume 2 are as follows:

- Removed all references to specific standards from Volume 2, in accordance with the status of Volume 7 as the definitive repository of standards data.

- Defined the terms Major Service Area (MSA), Mid-Level Service Area (MLSA), and Base Service Area (BSA).

- Harmonized the MLSAs and BSAs in Volume 7 with those in Volume 2, and incorporated the Volume 7 MLSA and BSA definitions into the Volume 2 definitions. In addition, certain MLSA definitions in Volume 2 that were not harmonized were adjusted to mirror the definitions of the MLSAs and BSAs in Volume 7.

- Modified the Technical Reference Model (TRM) shown in Figure 2-2 to depict the MLSAs that were in Volume 7 and are now incorporated in Volume 2, and identified that part of the model that contains the MSAs and MLSAs.

In addition, changes have been made to bring the guidance provided in this volume more in line with current policies. Work remains to be done to fully reflect the impact of the policy documents and decisions noted above; this edition of the TAFIM has been released to serve as a baseline and to make available throughout the DoD community the additions and modifications that have been implemented to date.

A historical perspective on the development of this volume and its changes over time appears in the Preface.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the AMWG will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is designated as the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.

- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.

- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web shortly after hard-copy publication. DISA is investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

This page intentionally left blank.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1:	Overview	3.0	30 April 1996
Volume 2:	Technical Reference Model	3.0	30 April 1996
Volume 3:	Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4:	DoD SBA Planning Guide	3.0	30 April 1996
Volume 5:	Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6:	DoD Goal Security Architecture	3.0	30 April 1996
Volume 7:	Adopted Information Technology Standards	3.0	30 April 1996
Volume 8:	HCI Style Guide	3.0	30 April 1996

Working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

PREFACE

The first draft of the Corporate Information Management (CIM) Technical Reference Model was submitted to DoD components for review on 4 September 1991. The review resulted in a number of editorial and minor technical changes that were included in Version 1.0 of the document. Additional comments and issues received as a result of staffing Version 1.0 resulted in the development of Version 1.1 of the document. Version 1.1 was submitted to the Information Technology Policy Board (ITPB) for approval in December 1991. On 12 February 1992, the Director of Defense Information (DDI) approved the use of Version 1.1 of the CIM Technical Reference Model by all DoD components. Version 1.1 was circulated widely within DoD and submitted to industry and other government activities for review. On 25 August 1992, the DDI approved the use of Version 1.2 of the Technical Reference Model.

The foremost issue identified during the review of the draft CIM Technical Reference Model was the extent to which future editions would expand to conform to either the National Institute of Standards and Technology (NIST) Application Portability Profile (APP) or emerging reference models based on the work of the Institute of Electrical and Electronics Engineers (IEEE), Open Software Foundation (OSF), UNIX International (UI), International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Joint Technical Committee 1 (JTC1), X/Open Limited, Accredited Standards Committee (ASC), and other national/international activities. The DISA JIEO staff continues to consult extensively with NIST and other external organizations on this issue.

The first major change to the CIM Technical Reference Model was the addition of the draft NIST FIPS Publication on the Government Network Management Profile (GNMP) that was added to the CIM standards profile as part of Version 1.1. Certain military features not yet contained in the draft GNMP Federal Information Processing Standard (FIPS) were to be documented in MIL-STD-2045-38000, Network Management for DoD Communications, and forwarded to NIST for incorporation in future versions of the FIPS. The recommendation to include the Draft GNMP FIPS and associated Military Standard was approved by the Architecture Methodology Working Group.

Version 1.2, known as the Technical Reference Model for Information Management, was published in May 1992. Version 1.2 included FIPS Publication 161 (Electronic Data Interchange), a limited discussion of Ada bindings, and a requirement for standards conformance testing. In addition, the figures in Version 1.2 were modified slightly in response to a number of vendor and DoD comments. Version 1.2 also contained a detailed discussion on security services and standards.

Version 1.3 of the Technical Reference Model was published in December 1992 and incorporated as Volume 3 of the Technical Architecture Framework for Information Management (TAFIM). Version 1.3 provided the necessary changes for closer alignment of the Technical Reference Model with the IEEE POSIX 1003.0 Draft Guide. Also included in Version 1.3 were the results of the harmonization of the Technical Reference Model and the

Computer-aided Acquisition and Logistics Support (CALS) architecture. Additional CALS standards, such as raster graphics, and an expanded discussion about each standard, to include up-to-date document references, were provided. In addition, several vendor comments submitted during the review of Version 1.1 were addressed.

Version 2.0 of the Technical Reference Model addressed several additional important topics. This version included the services and standards needed to support DoD's distributed computing requirements. In addition, requirements for internationalization services resulting from harmonization of the Technical Reference Model with North Atlantic Treaty Organization (NATO) reference model development efforts were addressed. Further, a new objective was added and the security material was significantly supplemented throughout the document to reflect the integration of the DoD Goal Security Architecture into the TAFIM domain. This version also reflected agreements reached between the DISA Center for Architecture and the DISA Center for Standards to incorporate the DoD Profile of Standards, which is the Adopted Information Technology Standards, (AITS) in a separate TAFIM volume, Volume 7. The DoD Profile of Standards or AITS, which corresponds to the reference model services and interfaces, is described in detail in Volume 7.

Areas to be addressed in future versions of Volume 2 include services and standards for tactical systems, imagery, and multimedia data transfer. A set of metrics, based on the NIST APP, will be provided to assist users in choosing extensions to the current standards in those areas where standards do not exist, or where consensus has not been achieved. Features and services required by the Communication-Electronics Accommodation Program (CAP) in support of handicapped access to DoD computer resources will also be added.

Version 3.0 of the Technical Reference Model addresses the harmonization of Volumes 2 and 7. In addition, the definitions of MSAs, MLSAs, and BSAs have been included in Volume 2. Further, Version 3.0 harmonized the MLSA and BSAs in Volume 7 with those Volume 2 and incorporated the Volume 7 MLSA definitions into Volume 2. The Technical Reference Model graphic was modified to depict the new MLSAs from Volume 7.

CONTENTS

FIGURES

This page intentionally left blank.

1.0 INTRODUCTION

1.1 BACKGROUND

On 16 November 1990, the Secretary of Defense directed the implementation of the Department of Defense (DoD) Corporate Information Management (CIM) initiative, hereafter known as the DoD Information Management initiative, to strengthen the DoD's ability to apply computing, telecommunications, and information management capabilities effectively in the accomplishment of the DoD mission. Transition of the DoD's present information systems and associated information technology resources to a communications and computing infrastructure based on the principles of open systems architecture and systems transparency is a key strategy for implementing the Department's Information Management initiative. The development of a technical reference model and the selection of associated standards are first steps toward executing this strategy.

1.2 PURPOSE AND OBJECTIVES

The purpose of the Technical Reference Model described in this document is to provide a common conceptual framework, and define a common vocabulary so that the diverse components within the DoD can better coordinate acquisition, development, and support of DoD information systems. The Technical Reference Model also provides a high-level representation of the information system domain showing major service areas. DoD Components are required to apply the model to increase commonality and interoperability across the DoD, as directed by the Director of Defense Information (DDI) Policy Memorandum of 12 February 1992, Subject: Open Systems Implementation and the Technical Reference Model. On 25 August 1992, the DDI approved the use of the first update to the Technical Reference Model for Information Management (Version 1.2) (see Appendix C).

The model is not a specific system architecture. Rather, it establishes a common vocabulary and defines a set of services and interfaces common to DoD information systems. The reference model and standards profile define the target technical environment for the acquisition, development, and support of DoD information systems.

The objectives to be achieved through application of the technical reference model presented in this document are as follows:

- Improve user productivity
- Improve development efficiency
- Improve portability and scalability
- Improve interoperability
- Promote vendor independence

- Reduce life-cycle costs
- Improve security
- Improve manageability.

1.3 STANDARDIZATION EFFORTS

NIST is currently pursuing the definition of an Open System Environment (OSE), which encompasses the functionality needed to provide interoperability, portability, and scalability of computerized applications across networks of heterogeneous hardware/software platforms. In April 1991, NIST published the first version of the Application Portability Profile (APP), which defines a reference model and outlines a suite of selected specifications (i.e., standards) that define the interfaces, services, protocols, and data formats for implementation of OSE within the U.S. Government. In June 1993, NIST published Version 2.0 of the APP as NIST Special Publication 500-210. The Technical Reference Model is adapted from the NIST model to meet the requirements of DoD and conforms to NIST recommendations wherever possible. As NIST continues to evolve the APP, changes will be considered for incorporation into the Technical Reference Model. As DoD requirements evolve, proposed changes to the APP will be forwarded to NIST. DISA will continue to work with NIST and other national and international standards organizations to ensure that the NIST APP and emerging standards meet or are compatible with the needs of DoD.

1.4 APPROACH

Major DoD component documents, including the DoD Intelligence Information System (DODIIS) Reference Model, were analyzed using the NIST APP as a baseline to derive the Technical Reference Model. The maturity, stability, completeness, and availability of standards for the service areas defined in the Technical Reference Model were then assessed. Where adequate standards were not available or multiple conflicting standards were contending for consensus, an issue was identified and an action plan was established.

The Technical Reference Model does not represent a final position, but is an evolutionary target. As technology continues to advance and additional standards emerge, the Architecture Methodology Working Group will continue to update the standards profile and recommend refinements in the Reference Model to the Office of the Secretary of Defense (OSD).

1.5 DOCUMENT ORGANIZATION

The Technical Reference Model document consists of two sections and four appendices. Section 2 provides an overview of the Technical Reference Model, the principles upon which the model is based, and the services to be provided. References and acronyms are identified in Appendices A and B, respectively. Appendix C contains the DDI memoranda dated 12 February 1992 and 25 August 1992 concerning the Technical Reference Model. Appendix D contains instructions and a template for commenting on this document.

2.0 DOD TECHNICAL REFERENCE MODEL

2.1 OVERVIEW

Within the context of information systems, a reference model is defined to be a generally accepted representation that allows people to agree on definitions, build common understanding, and identify issues for resolution. A technical reference model is necessary to establish a context for understanding how the disparate technologies required to implement information management relate to each other. The model also provides a mechanism for identifying the key issues associated with applications portability, scalability, and interoperability. The Technical Reference Model is not a specific system design. Rather, it establishes a common vocabulary and defines a set of services and interfaces common to DoD information systems. The Technical Reference Model will serve to facilitate interoperability between mission-area applications, portability across mission areas, and cost reductions through the use of common services. The development and acceptance of the Technical Reference Model is critical to the successful implementation of the DoD Information Management initiative.

2.2 PRINCIPLES

The Technical Reference Model was devised to permit the DoD to take advantage of the benefits of open systems and the new technologies available in the commercial market. DoD-wide application of the model should result in cost savings over the long term. Section 1 outlined the Technical Reference Model objectives. The principles that support these objectives and that will be used to refine and implement the Reference Model are described below.

OBJECTIVE 1: IMPROVE USER PRODUCTIVITY

User productivity improvements will be realized by applying the following principles:

- **Consistent User Interface.** A consistent user interface will ensure that all user accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site. This has the benefits of simplifying training, facilitating the development of future applications, improving ease of use across applications, and promoting application portability.

- **Integrated Applications.** Applications available to the user will behave in a logically consistent manner across user environments. Support applications, such as office automation and electronic mail, will be used as an integrated set with mission area specific applications.

- **Data Sharing.** Databases will be shared across DoD in the context of security and operational considerations. Concepts and tools that promote data sharing include adherence to standard database development rules, the use of DoD data dictionary and software reuse libraries, and strong DoD commitment to resource sharing.

OBJECTIVE 2: IMPROVE DEVELOPMENT EFFICIENCY

The efficiency of development efforts will be improved by applying the following principles:

- **Common Development.** Applications that are common to multiple mission areas will be centrally developed or acquired.

- **Common Open Systems Environment.** A standards-based common operating environment, which accommodates the injection of new standards, technologies, and applications on a DoD-wide basis, will be established. This standards-based environment will provide the basis for development of common applications and facilitate software reuse.

- **Use of Products.** To the extent possible, hardware-independent, nondevelopmental items (NDI) should be used to satisfy requirements in order to reduce the dependence on custom developments and to reduce development and maintenance costs.

- **Software Reuse.** For those applications that must be custom developed, incorporating software reuse into the development methodology will reduce the amount of software developed and add to the inventory of software suitable for reuse by other systems.

- **Resource Sharing.** Data processing resources (hardware, software, and data) will be shared by all users requiring the services of those resources. Resource sharing will be accomplished in the context of security and operational considerations.

OBJECTIVE 3: IMPROVE PORTABILITY AND SCALABILITY

The portability and scalability of applications will be improved by applying the following principles:

- **Portability.** Applications that implement the model's paradigms will be portable, allowing for movement across heterogeneous computing platforms with minimal or no modifications. With portable applications, implementing activities will be able to upgrade their hardware base as technological improvements occur, with minimal impact on operations.

- **Scalability.** Applications that conform to the model will be configurable, allowing operation on the full spectrum of platforms depending on user requirements.

OBJECTIVE 4: IMPROVE INTEROPERABILITY

Interoperability improvements across applications and mission areas can be realized by applying the following principles:

- **Common Infrastructure**. The DoD will develop and implement a communications and computing infrastructure based on open systems and systems transparency including, but not limited to, operating systems, database management, data interchange, network services, network management, and user interfaces. The basis for common infrastructure is to identify core capabilities having a commonality of application across services. This, in the near term, enables the migration from static and monolithic applications (stovepipes) to a more open environment enabling data and data format transparency across heterogeneous platforms.

- **Standardization.** By implementing standards from the DoD Profile of Standards (see Section 3), applications will be provided and will be able to use a common set of services that improve the opportunities for interoperability.

OBJECTIVE 5: PROMOTE VENDOR INDEPENDENCE

Vendor independence will be promoted by applying the following principles:

- **Interchangeable Components.** Hardware and software supporting or migrating to open systems compliance will be acquired or implemented, so that upgrades or the insertion of new products will result in minimal disruption to the user's environment.

- **Non-Proprietary Specifications.** Capabilities will be defined in terms of non-proprietary specifications that support full and open competition and are available to any vendor for use in developing commercial products.

OBJECTIVE 6: REDUCE LIFE-CYCLE COSTS

Life-cycle costs can be reduced by applying most of the principles discussed above. In addition, the following principles directly address reducing life-cycle costs:

- **Reduced Duplication.** Replacement of "stovepipe" systems and "islands of automation" with interconnected open systems, which can share data and other resources, will dramatically reduce overlapping functionality, data duplication, and unneeded redundancy.

- **Reduced Software Maintenance Costs.** Software complexity may increase with increased user demand for services such as distributed processing and distributed database services. However, if the principles described above are implemented, reductions in software maintenance will be realized because there will be less software to maintain. In those cases where the number of DoD users is small, increased use of standard nondevelopmental software will further reduce costs since vendors of such software distribute their product maintenance costs across a much larger user base.

- **Reduced Training Costs.** A reduction in training costs will be realized because users rotating to new organizations will already be familiar with the common systems and consistent human computer interfaces (HCI).

OBJECTIVE 7: IMPROVE SECURITY

Security will be improved in DoD information systems by satisfaction of the following principles for information systems that may need to operate simultaneously in various DoD environments (tactical, strategic, and sustaining base):

- **Uniform Security Accreditation and Certification.** Uniform certification and accreditation procedures will not only reduce the time needed to approve system operation but will result in more consistent use of security mechanisms to protect sensitive data.

- **Consistent Security Interfaces.** Consistent security interfaces and labeling procedures will reduce errors when managing sensitive data and reduce learning time when changing from system to system. Not all mission-area applications will need the same suite of security features, but any features used will be consistent across applications. Users will see the same security labels in a common format and manage them in the same way.

- **Support for Simultaneous Processing in Single Platforms of Different Information Domains.** Security protection will be provided for simultaneous processing of various categories of information within a single system. Information systems that can support multiple security policies can support multiple missions with varying sensitivity and rules for protected use. This will include support of simultaneous processing under multiple security policies of any complexity or type, including policies for sensitive unclassified information and multiple categories of classified information. This type of support will also permit users with different security attributes to simultaneously use the system. Separate or dedicated information systems for processing information controlled by different security policies will be reduced or eliminated.

- **Support for Simultaneous Processing in a Distributed System of Different Information Domains.** Security protection will be provided for simultaneous processing of various categories of information in a distributed environment. This protection will apply to processing of information controlled by multiple security policies in distributed networks using heterogeneous platforms and communications networks. This will greatly extend the flexibility of the system implementor in providing cost-effective information systems based on open systems principles.

- **Support for Use of Common User Communications Systems.** Security protection will be provided in such a way as to permit use of common carrier (public) systems for communications connectivity. It will also permit the use of Department-owned common user communications systems. This use of public and Department common user global communications networks will result in the potential for enhanced cost effective interoperability across mission areas.

OBJECTIVE 8: IMPROVE MANAGEABILITY

Management improvement can be realized by applying the following principles:

- **Consistent Management Interface.** Consistency of management practices and procedures will facilitate management across all applications and their underlying support structures. Users will accomplish work more efficiently by having the management burden simplified through such an interface.

- **Management Standardization.** By standardizing management practices, control of individual and consolidated processes will be improved in all interoperable scenarios.

- **Reduced Operation, Administration, and Maintenance (OA&M) Costs.** OA&M costs will be reduced through the availability of improved management products and increased standardization of objects being managed.

2.3 GENERIC DOD TECHNICAL REFERENCE MODEL

The generic DoD Technical Reference Model is a set of concepts, entities, interfaces, and diagrams that provides a basis for the specification of standards. To a large extent, the Technical Reference Model adopts the foundation work of the IEEE POSIX P1003.0 Working Group as reflected in their Draft Guide to the POSIX Open System Environment (POSIX.0). The POSIX Guide has reached a degree of maturity such that it is undergoing the IEEE balloting process to be sanctioned as an official IEEE document. Within the guide, an interface is defined as "a shared boundary between the two functional units." The functional units are referred to as "entities" when discussing the classification of items related to application portability.

The basic elements of the generic DoD Technical Reference Model are those identified in the POSIX Open System Reference Model and are presented in Figure 2-1. As shown in the figure, the model includes three classes of entities and two types of interfaces as follows:

- Application Software Entity

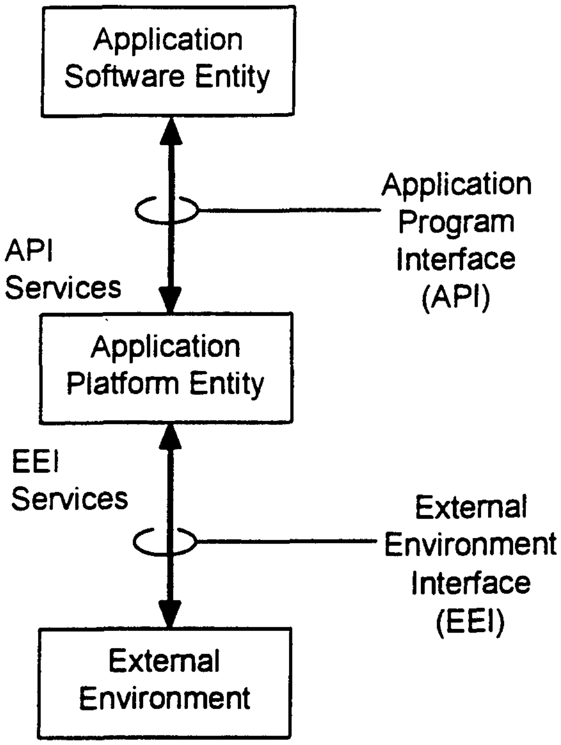
- Application Program Interface (API)

- Application Platform Entity

- External Environment Interface (EEI)

- External Environment.

This model has been generalized to such a degree that it can accommodate a wide variety of general and special purpose systems. More detailed information is presented in subsequent sections; however, the service specifications allow for subsets or extensions as needed.


Application Software Entity
Application Program Interface (API)
API Services
Application Platform Entity
EEI Services
External Environment Interface (EEI)
External Environment

Reference: IEEE Draft Guide to the POSIX Open System Environment, June 1992

Figure 2-1. Generic DoD Technical Reference Model

From the perspective of the application software entity, these services are provided by an application platform whether the particular services are provided from the local platform or from remote platforms that may comprise one or more nodes of a larger distributed system. Volume 3 of the TAFIM explains how this generic model can be applied in a distributed environment.

2.3.1 Application Software Entity

In the past, custom systems were developed for specific hardware platforms using proprietary systems software (e.g., operating system, text editor, file management utilities). Such customization was necessary because Government requirements were often more localized than those of the commercial marketplace. These systems were not designed to interoperate with other systems nor to be portable to other hardware platforms. In addition, different systems were developed to perform similar functions at different levels of the overall DoD organization (national, theater, and unit) and for the different Services, (Army, Navy, Air Force, Marine Corps). As a result, many of the systems that were developed included functions redundant with those of other applications. This situation often hindered systems evolution toward greater interoperability, data sharing, portability, and software reuse.

The Technical Reference Model promotes the goals of developing modular applications and promoting software reuse to support the broad range of activities that are integral to any organization. To satisfy these goals, functional (mission-area) applications development will, in many respects, become an integration activity as much as a development activity. Application development will likely be accomplished by dividing and/or consolidating common functional requirements into discrete modules. Previously developed reusable code or Government-off-the-shelf (GOTS) applications that could satisfy some, if not all, of the new functional requirements would be identified. Such reusable code/applications would then be integrated, to the extent possible, to become the software pieces necessary to complete the mission and/or support applications that will satisfy all of the requirements.

In the Technical Reference Model, applications are divided into mission area applications and support applications. A common set of support applications forms the basis for the development of mission-area applications. Mission-area applications should be designed and developed to access this set of common support applications. As explained in Volume 3, APIs are also used to define the interfaces between mission-area applications and support applications.

The *DoD Goal Security Architecture* (DGSA) in Volume 6 also anticipates the expanding use of NDI products in information system implementations. For this reason, two categories of software are identified, trusted and untrusted. Both categories may have been acquired for an information system implementation as NDI products. However, the trusted software will have been evaluated in accordance with criteria established by responsible agencies for information system security and will need to be maintained under strict configuration management control. Trusted software will mediate the access of all untrusted software to information system resources. Such control, which the DGSA suggests should be in the operating system kernel, will provide the necessary security protection by maintaining separation among applications at different security levels that are simultaneously processing.

2.3.2 Application Program Interface

The API is defined as the interface between the application software and the application platform across which all services are provided. It is defined primarily in support of application portability, but system and application software interoperability also are supported via the communication services API and the information services API. The API specifies a complete interface between the application and the underlying application platform and may be divided into the following groups:

- System Services API (including APIs for Software Engineering Services and Operating System Services)

- Communications Services API (including APIs for Network Services)

- Information Services API (including APIs for Data Management Services and Data Interchange Services)

- Human/Computer Interaction Services API (including APIs for User Interface Services and Graphics Services).

The first API group, System Services, is required to provide access to services associated with the application platform internal resources. The last three API groups (Communications

Services, Information Services, and Human/Computer Interaction Services) are required to provide the application software with access to services associated with each of the external environment entities. APIs for services that cut across the areas are included among all groups where applicable.

A standardized API should be used for accessing security mechanisms. The use of the operating system kernel for maintaining separation among processes executing at different security levels means that this API would be included in the System Services API category above. Such an API will promote independence of security services and security mechanisms, offering transparency to users and applications. This independence will allow different security mechanisms to be accommodated at various stages in an information system life cycle.

2.3.3 Application Platform Entity

The Application Platform is defined as the set of resources that support the services on which application software will execute. It provides services at its interfaces that, as much as possible, make the implementation-specific characteristics of the platform transparent to the application software.

To assure system integrity and consistency, application software entities competing for application platform resources must access all resources via service requests across the API. Examples of application platform services may include an operating system kernel, a realtime monitor program, and all hardware and peripheral drivers.

The application platform concept does not imply or constrain any specific implementation beyond the basic requirement to supply services at the interfaces. For example, the platform might be a single processor shared by a group of applications, a multiprocessor at a single node, or it might be a large distributed system with each application dedicated to a single processor.

The application platform implementations that use the Technical Reference Model may differ greatly depending upon the requirements of the system and its intended use. It is expected that application platforms defined to be consistent with the Technical Reference Model will not necessarily provide all the features discussed here, but will use tailored subsets for a particular set of application software.

2.3.4 External Environment Interface

The External Environment Interface (EEI) is the interface between the application platform and the external environment across which information is exchanged. It is defined primarily in support of system and application software interoperability. User and data portability are directly provided by the EEI, but application software portability also is indirectly supported by reference to common concepts linking specifications at both API and EEI. The EEI specifies a complete interface between the application platform and the underlying external environment, and may be divided into the following groups:

- Human/Computer Interaction Services EEI

- Information Services EEI

- Communications Services EEI.

The Human/Computer Interaction (HCI) Services EEI is the boundary across which physical interaction between the human being and the application platform takes place. Examples of this type of interface include CRT displays, keyboards, mice, and audio input/output devices. Standardization at this interface will allow users to access the services of compliant systems without costly retraining.

The Information Services EEI defines a boundary across which external, persistent storage service is provided, where only the format and syntax are required to be specified for data portability and interoperability.

The Communications Services EEI provides access to services for interaction between application software entities and entities external to the application platform, such as application software entities on other application platforms, external data transport facilities, and devices. The services provided are those where protocol state, syntax, and format all must be standardized for application interoperability.

Security mechanisms to provide for security services in EEIs will be implemented similarly to those required for communications among distributed platforms. That is, the EEIs facilitate communications among distributed platforms. Such implementations will occur primarily in the cross-platform service areas of security and system management. See Sections 2.4.4.2 and 2.4.4.3.

2.3.5 External Environment

The External Environment contains the external entities with which the application platform exchanges information. These entities are classified into the general categories of human users, information interchange entities, and communications entities. Human users are not further classified, but are treated as an abstract, or average person. Information interchange entities include, for example, removable disk packs and floppy disks. Communications entities include telephone lines, local area networks, cabling, and packet switching equipment.

Doctrinal mechanisms (physical, administrative, and personnel) will provide for required security protection of information system components in the external environment.

2.4 DETAILED DOD TECHNICAL REFERENCE MODEL

Figure 2-2 expands upon Figure 2-1 to present the DoD Technical Reference Model entities and interfaces, including the service areas of the Application Platform and related services. Figure 2-2 only depicts entities, interfaces, and service areas and does not imply interrelationships among the service areas.

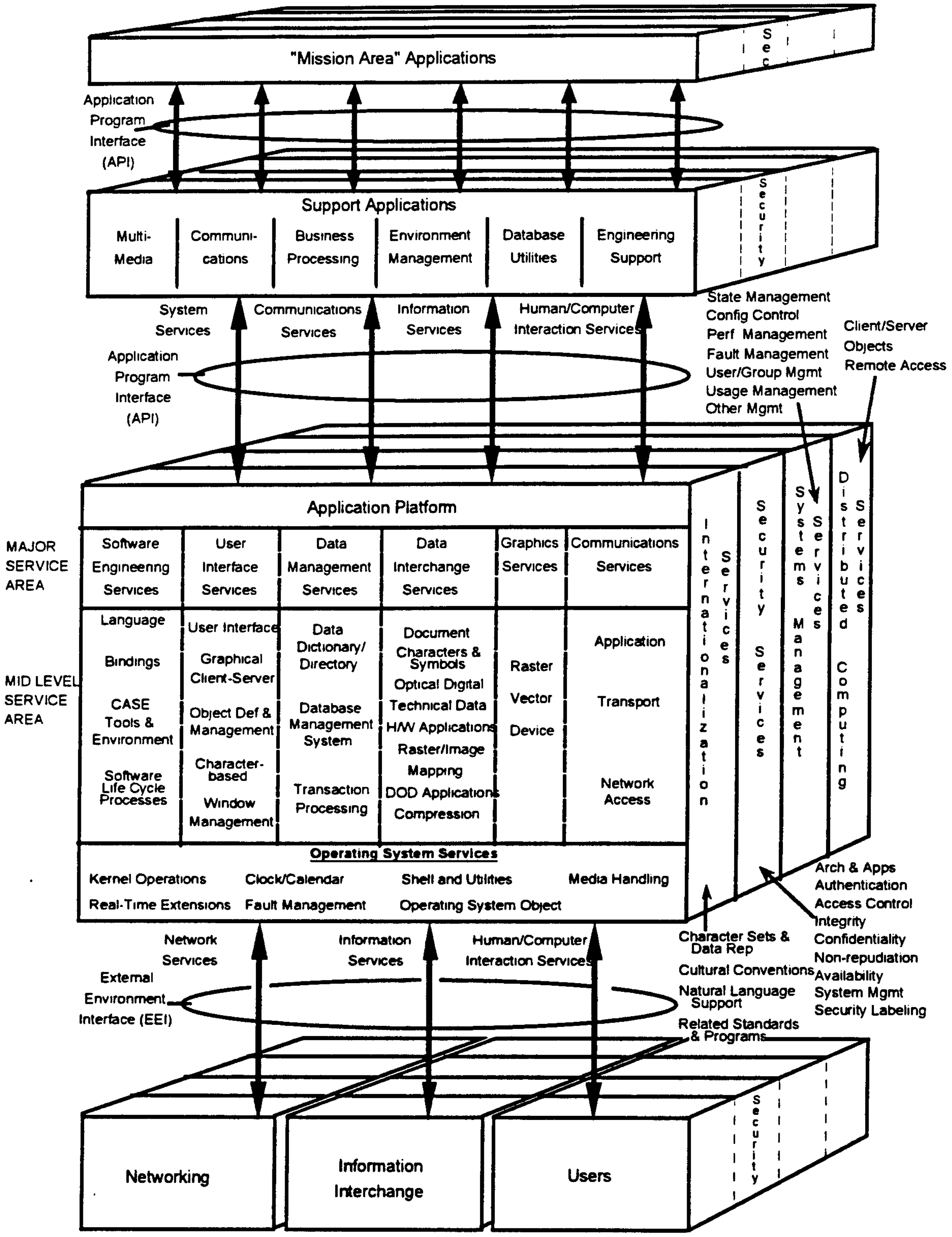


Figure 2-2. Detailed DoD Technical Reference Model

Users should assess their own requirements and create a profile of services, interfaces, and standards that satisfy their own mission-area needs. Users who have adopted earlier versions of the figure should consider adopting the new version of the figure only when planning a major revision of their documentation.

2.4.1 Mission Area Applications

Mission area applications implement specific end-user requirements or needs (e.g., payroll, accounting, materiel management, personnel, control of real-time systems, analysis of order of battle). This application software may be COTS or GOTS, custom developed, or a combination of these. In addition to application software, an information system includes data that can be application specific (e.g., a log of invoices and payments) or an integral part of the software (e.g., application parameters, screen definitions, diagnostic messages). Information systems also include training (e.g., tutorials and on-line help), support tools (e.g., programs for software development, self-test diagnostics), and system management aids (e.g., system administration).

2.4.2 Support Applications

Support applications are common applications (e.g., E-mail, word processing, spreadsheets) that can be standardized across individual or multiple mission areas. The services they provide can be used to develop mission-area-specific applications or can be made available to the user. Support applications may be COTS products selected to provide a service in a common manner, or they may be GOTS applications developed to meet a DoD-unique need and reused in multiple information systems. The Defense Information Infrastructure (DII) Common Operating Environment (COE) includes several support applications to provide common functions such as message handling, network browsing, and mapping. For example, the Joint Mapping Toolkit (JMTK) provides objects and services to support geospatial analysis, mapping (visual) display, geospatial database management, and image preprocessing.

The set of services described in this section provides initial capabilities that will be used to define, acquire, and develop common, shared applications. The services have been grouped into categories by function. The categories and list of services will most likely change over time. New services will be added, or in some cases, existing services will be rearranged and merged into new categories. Some of the services, particularly those found in the multimedia category, will be used as building blocks to implement other services. An implementation of a support application may actually merge several services from several different categories.

The combination of support applications with the services of the platform layer provides the basis for a "common operating environment" to support mission applications. The DII COE implements this concept with a precisely defined client/server architecture for how system components fit together; a standard extensible run-time operating environment that includes "look and feel" operating system and windowing environment standards; a clearly defined set of already implemented, reusable functions; and a collection of APIs for accessing COE components.

2.4.2.1 Multimedia

Multimedia services provide the capability to manipulate and manage information consisting of text, graphics, images, video, and audio. These services can be used directly by mission area applications, but, they can also be used by other support applications to satisfy a common requirement. Multimedia services include:

- **Text processing** services, including the capability to create, edit, merge, and format text.
- **Document processing** services, including the capability to create, edit, merge, and format documents. These services enable the composition of documents that incorporate graphics, images, and even voice annotation, along with stylized text. Included are advanced formatting and editing services such as style guides, spell checking, use of multiple columns, table of contents generation, headers and footers, outlining tools, and support for scanning images into bit-mapped formats.
- **Electronic publishing** services, including incorporation of photographic quality images and color graphics, and advanced formatting and style features such as wrapping text around graphic objects or pictures and kerning (i.e., changing the spacing between text characters). These services also interface with sophisticated printing and production equipment.
- **Geographic information system (GIS)** services, including the capability to create, combine, manipulate, analyze, and present geospatial information. This includes the creation of entity symbology that overlays the map background display and access to standard symbol libraries.
- **Image processing** services providing for the capture, scan, creation, and edit of images in accordance with recognized image formatting standards.
- **Video processing** services, including the capability to capture, compose, and edit video information. Still graphics and title generation services are also provided.
- **Audio processing** services, including the capability to capture, compose, and edit audio information.
- **Multimedia processing** services, including the capability to compress, store, retrieve, modify, sort, search, and print all or any combination of the above-mentioned media, and to perform these actions on two or more types of media simultaneously. This includes support for microform media, optical storage technology that allows for storage of scanned or computer produced documents using digital storage techniques, a scanning capability, and data compression. Additionally, multimedia processing includes hypermedia processing. Hypermedia provides the capability to create and browse documents that allow users to interactively navigate through the document using information embedded in the document.

2.4.2.2 Communications

Communications services provide the capability to send, receive, forward, and manage electronic and voice messages. They also provide real-time information exchange services in support of interpersonal conferences. These services include:

- **Personal messaging** services, including the capability to send, receive, forward, store, display, and manage personal messages. This includes the capability to append files and documents to messages. Messages may include any combination of data, text, audio, graphics, and images and should be capable of being formatted into standard data interchange formats. This service includes the use of directories and distribution lists for routing information, the ability to assign priorities, the use of pre-formatted electronic forms, and the capability to trace the status of messages. Associated services include a summarized listing of incoming messages, a log of messages received and read, the ability to file or print messages, and the ability to reply to or forward messages.

- **Organizational messaging** services, including the capability to send, receive, forward, display, retrieve, prioritize, and manage predefined and unformatted organizational messages. Organizational messages should use standard data interchange formats and may include any combination of data, text, audio, graphics, and images. This includes the capability to review and authenticate messages. Incoming message processing services include receipt, validation, distribution, and dissemination of incoming unformatted messages based on message profiling, message precedence, and system security restrictions. User support services include the selection and display of messages from a message queue, on-line management of search profiles, search and retrieval of stored messages based on message content comparison to queries formulated by the analysts, and composition of record messages for transmission. Outgoing message processing services include coordination by the command's staff organizations, authorized release, and verification of record messages prior to transmission.

- **Enhanced telephony** services, including call forwarding, call waiting, programmed directories, teleconferencing, automatic call distribution (useful for busy customer service areas), call detail recording, and voice mail.

- **Shared screen teleconferencing** services that allow two or more users to communicate and collaborate using audio teleconferencing with common "shared" workstation windows that refresh whenever someone displays new material or changes an existing display. Every user is provided the capability to graphically annotate or modify the shared conference window.

- **Video teleconferencing** services that provide two-way video transmission between different sites. These services include full motion display of events and participants in a bi-directional manner, support for the management of directing the cameras, ranging from fixed position, to sender directed, to receiver directed, to automated sound pickup.

- **Broadcast** services that provide one-way audio or audio/video communications services between a sending location and multiple receiving locations.

- **Computer conferencing** services that allow groups to participate in conferences via computer workstations. These conferences may not occur in real time. Conferees or invited guests can drop in or out of conferences or subconferences at will. The ability to trace the exchanges is provided. Services include exchange of documents, conference management, recording facilities, and search and retrieval capabilities.

2.4.2.3 Business Processing

Business support services provide common office functions used in day-to-day operations. Business support services include:

- **Spreadsheet** services, including the capability to create, manipulate, and present information in tables or charts. This capability should include fourth-generation-language-like capabilities that enable the use of programming logic within spreadsheets.

- **Project management** services, including tools that support the planning, administration, and management of projects.

- **Calculation** services, including the capability to perform routine and complex arithmetic calculations.

- **Calendar** services, including the capability to manage personal tasks and time and to coordinate multiple personal schedules via an automated calendar.

2.4.2.4 Environment Management

This type of service is broader in scope than the other categories in that it exists primarily to manage a particular data processing and/or communications environment. Environment management services integrate and manage the execution of platform services for particular applications and users. These services are invoked via an easy-to-use, high-level interface that enables users and applications to invoke platform services without having to know the details of the technical environment. The environment management service determines which platform service is used to satisfy the request and manages access to it through the API.

- **Batch processing** services support the capability to queue work (jobs) and manage the sequencing of processing based on job control commands and lists of data. These services also include support for the management of the output of batch processing, which frequently includes updated files or databases and information products such as printed reports or electronic documents. Batch processing is performed asynchronously from the user requesting the job.

- **Transaction processing** services provide support for the on-line capture and processing of information in an interactive exchange with the user. This typically involves predetermined sequences of data entry, validation, display, and update or inquiry against a file or database. It also includes services to prioritize and track transactions. Transaction processing services may include support for distribution of transactions to a combination of local and remote processors.

- **Information presentation and distribution** services are used to manage the distribution and presentation of information from batch and interactive applications. These services are used to shield mission-area applications from how information is used. They allow mission area applications to create generic pools of information without embedding controls that dictate the use of that information. Information distribution and presentation services include the selection of the appropriate formatting services required to accomplish the distribution and presentation of information to a variety of mission-area applications, support applications, and users. It also includes the capability to store, archive, prioritize, restrict, and recreate information.

- **Computer-based training** services provide for integrated training environment on user workstations. Training is available on an as-needed basis for any application available in the environment. Electronic messages are provided at the stroke of a key from anywhere within the application. This includes tutorial training on the application in use and the availability of off-line, on-site interactive training. The DoD on-line training environment will provide in-depth training to the new user, guidance to the novice user, and refresher material for the more experienced user. Computer-based training includes on-line documentation services. As a system service, generalized Help Files that have index, contents, and context-sensitive definitions must be added to all applications. The goal is for a user, through a system-managed activity, to be able to obtain help at any point, while on line.

2.4.2.5 Database Utilities

Database utility services provide the capability to retrieve, organize, and manipulate data extracted from a database management system. These common services provide a consistent interface to the user while providing access to a variety of databases. Database utility services include:

- **Query processing** services that provide for interactive selection, extraction, and formatting of stored information from files and databases. Query processing services are invoked via user-oriented languages and tools (often referred to as fourth-generation languages), which simplify the definition of searching criteria and aid in creating effective presentation of the retrieved information (including use of graphics). Fourth-generation languages are generally all proprietary. Some are in the public domain (for example, Dbase clones are generally referred to as "Xbase" systems), but these all started as proprietary systems. As yet, no public domain fourth-generation language is in wide business use.

- **Screen generation** services that provide the capability to define and generate screens that support the retrieval, presentation, and update of data.

- **Report generation** services that provide the capability to define and generate hardcopy reports composed of data extracted from a database.

- **Networking/concurrent access services** that manage concurrent user access to database management system (DBMS) services.

2.4.2.6 Engineering Support

Engineering support services include support for analysis, design, modeling, development, and simulation for a wide variety of users and environments. This includes computer-aided design services for designing, drafting, and producing engineering drawings. It also includes services provided by decision support development tools and expert system shells.

- **Computer-aided design (CAD)** services provide high-precision drawing tools and modeling capabilities to allow production of engineering specification drawings and other precise drawings.

- **Decision support** services provide interactive modeling and simulation tools that support analysis of alternative decisions.

- **Expert system** services provide artificial intelligence capabilities usually based on knowledge- or rules-based inference engines that recommend or take actions based on presented situations and prior "experiences."

- **Modeling and simulation** services provide the capability to capture or set object characteristics or attributes and parameters of a system of objects, and to portray the relationships and interactions of the objects to assist in the analysis of the system.

2.4.3 Application Platform Service Areas

This section provides a characterization of the terms used to describe the Application Platform Service Areas of the Technical Reference Model (TRM). These terms provide a common definition for the services and interfaces used by DoD information systems and apply to all volumes of the TAFIM. The TAFIM describes the information technology (IT) services provided by the Application Platform Service Area in three levels of detail: Major Service Area, Mid-Level Service Area, and Base Service Area.

Each major heading (MSA) establishes a grouping of services or functionality defined by industry standards and is expressed in a way to be consistent with the manner in which the standards bodies are addressing these groups. The sub-headings, (MLSA and BSA) identify more specific, concrete examples of the functionality represented by the major grouping.

The functionality described by the MSAs, MLSAs, and BSAs defines the services available from the Application Platform across the platform interfaces (APIs and EEIs). The MSAs and MLSAs are identified in the Application Platform Service Area of the TRM, while the BSAs are addressed in Volume 7.

Major Service Area: The Major Service Area category is the highest level of IT functionality. MSAs provide the overall set of standard services that support the objectives of application portability and system interoperability. The MSAs include Software Engineering Services, User Interface Services, Data Management Services, Data Interchange Services, Graphics Services, and Network Services.

Mid-Level Service Area: MSAs are divided into areas, called Mid-Level Service Areas, that provide like functionality and further decompose the IT functionality. This decomposition is intended to provide a more precise description of each MSA. The MLSAs are represented under the MSAs in bold. The number of categories in each MLSA varies, depending on the variation and complexity of the functionality included in the MSA. The MSAs and MLSAs are fully described in the following sections, 2.4.3.1 through 2.4.3.7.

Base Service Area: The BSA is the next level of granularity below the Mid-Level Service Area and provides the most precise description of IT functionality in a Major Service Area. The BSAs further decompose the IT functionality in each MLSA category. The number of BSAs for any MLSA will vary depending on the complexity of the functionality covered by the MLSA category. The BSAs are fully described in the Information Technology Standards Guidance (ITSG), which supports the development of Volume 7.

2.4.3.1 Software Engineering Services

Professional system developers require tools appropriate to the development and maintenance of applications. These capabilities are provided by software engineering services, which include:

- **Language** services provide the basic syntax and semantic definition for use by a software developer to describe the desired application software function. Shell and executive script language services enable the use of operating system commands or utilities rather than a programming language. Shells and executive scripts are typically interpreted rather than compiled, but some operating systems support compilers for executive scripts. Other programming tools may use procedural or object-oriented languages to define the functionality of the desired applications. Third-generation languages provide primarily command line interfaces and text-based code for defining the applications, while more recent fourth-generation languages are forms-based and provide a graphical interface.

- **Bindings** and object code linking provide the ability for programs to access the underlying application and operating system platform through APIs that have been defined independently of the computer language. They are used by programmers to gain access to these services using methods consistent with the operating system and specific language used. Only Ada refers to such actions as "language bindings." All other compilers, DBMSs, and system software refer to such actions as "linking." Linking is operating system dependent, but language independent.

- **Computer-Aided Software Engineering (CASE) tools and environment** include systems and programs that assist in the automated development and maintenance of software. These include, but are not limited to, tools for requirements specification and analysis, for design work and analysis, for creating and testing program code, for documenting, for prototyping, and for group communication. The interfaces among these tools include services for storing and retrieving information about systems and exchanging this information among the various components of the system development environment. An adjunct to these capabilities is the ability to manage and control the configuration of software components, test data, and libraries. Other fourth-generation language tools include software development tools such as artificial intelligence tools and the UNIX command "imake."

- **Software life cycle processes** identify distinct phases in the software life cycle, which is the period of time that begins when a software product is conceptualized and ends when the software is no longer available for use. It includes a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals). The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and the retirement phase.

2.4.3.2 User Interface Services

User interface services define how users may interact with an application. Depending on the capabilities required by users and the applications, these interfaces may include the following specifications:

- **User interface** services define how users may interact with an application. They provide a consistent way for people who develop, administer, and use a system to gain access to applications programs, operating systems, and various system utilities. The user interface is a combination of menus, screen design, keyboard commands, command language, and help screens, which create the way a user interacts with a computer. The use of mice, touch screens. and other input hardware are included as part of the user interface.

- **Graphical client-server operations** define the relationships between client and server processes operating within a network, in particular, graphical user interface display processes. In this case, the program that controls each display unit is a server process, while independent user programs are client processes that request display services from the server.

- **Object definition and management** services define characteristics of display elements such as color, shape, size, movement, graphics context, user preferences, and interactions among display elements.

- **Character-based user interface** can be either a command-line interface or a menu-driven interface similar to a graphical user interface, but it does not use graphics and may depend solely on the keyboard for user input, i.e., not make use of an explicit pointing device.

Modern systems and applications are and will continue to be based upon graphical user interfaces and the associated standards for such systems. However, many legacy systems still include a large number of character-based terminals and interfaces.

- **Window management** specifications that define how windows are created, moved, stored, retrieved, removed, and related to each other.

User interfaces are often the most complex part of system development and maintenance. Volume 8, the *DoD Human-Computer Interface (HCI) Style Guide*, provides a common framework to document and define functional goals, objectives and requirements, and provides guidance to assist DoD application designers in implementing HCI style standards. Within the past few years, significant advances have been made in user interfaces, both in ease of use and in reducing the development effort required. Although other technologies can be used, most users think of a user interface in terms of a graphical user interface (GUI). A GUI allows a user to specify actions by dragging and dropping or pointing and clicking on an icon that is a pictorial metaphor for the object being acted upon. A GUI can also depict several actions simultaneously by presenting multiple windows.

The services associated with a windows system include the visual display of information on a screen that contains one or more windows or panels, support for pointing to an object on the screen using a pointing device such as a mouse or touch-screen, and the manipulation of a set of objects on the screen through the pointing device or through keyboard entry.

2.4.3.3 Data Management Services

Central to most systems is the management of data that can be defined independently of the processes that create or use it, maintained indefinitely, and shared among many processes. Data management services include:

- **Data dictionary/directory** services allow data administrators and information engineers to access and modify data about data (i.e., metadata). Such data may include internal and external formats, integrity and security rules, and location within a distributed system. Data dictionary/directory services also allow end users/applications to define and obtain data that are available in the database. Data administration defines the standardization and registration of individual data element types to meet the requirements for data sharing and interoperability among information systems throughout the enterprise. Data administration functions include procedures, guidelines, and methods for effective data planning, analysis, standards, modeling, configuration management, storage, retrieval, protection, validation, and documentation.

- **Database management system** services provide data administration, managed objects functionality, and controlled access to and modification of structured data. To manage the data, the DBMS provides concurrency control and facilities to combine data from different schemas. Facilities may also include the capability to manage data in a distributed computing environment where data is stored on multiple, heterogeneous platforms. DBMS services are accessible through a programming language interface, an interactive data manipulation language interface such as SQL, or an interactive/fourth-generation language interface. For efficiency, database management systems generally provide specific services to create, populate, move, backup, restore/recover, and archive databases, although some of these services could be provided by general file management capabilities described in operating system services.

- **Transaction processing** services support the definition and processing of "transactions." A transaction is a "unit of work" consisting of a series of operations that must be completed together. A transaction is characterized by the ACID properties:

  - Atomicity: implies that the operations of work are either all performed, or none of them are performed

  - Consistency: implies that the operations of a unit of work, if performed at all, are performed accurately, correctly, and with validity, with respect to applications semantics

  - Isolation: implies that the partial results of a unit of work are not accessible, except by operations which are part of the unit of work, and also implies that units of work which share bound data can be serialized

  - Durability: implies that all the effects of a completed unit of work are not altered by any sort of failure. While transaction processing is often associated with database management, it is also applicable in operating systems and communications, as well as physical actions (e.g., dispensing money at a cash machine) that are unrelated to database management.

2.4.3.4 Data Interchange Services

Data interchange services provide specialized support for the interchange of information between applications and to/from the external environment. These services are designed to handle data interchange between applications on the same platform and applications on different (heterogeneous) platforms.

- **Document interchange** services are supported by specifications for encoding the data (e.g., text, pictures, numerics, special characters) and both the logical and visual structures of electronic documents. Services support document exchange between heterogeneous computer systems, exchange of military formatted messages, and electronic forms interchange.

- **Characters and symbols** services provide for interchange of character sets and fonts and standardized date and time representation.

- **Optical digital technologies** (ODT) represents technologies that use the reflective properties of light and an optical recording surface to capture, encode, decode, and store data. ODT predominantly encompasses optical media, optical drives, and scanners.

- **Technical data interchange** services provide facilities for the exchange of technical data. This includes standards for the interchange of graphics data, typically vector graphics, technical specifications, and product data. Product data encompasses technical drawings, documentation, and other data required for product design and manufacturing, including geometric and nongeometric data such as form features, tolerances, material properties, and surfaces.

- **Hardware applications** services provide data interchange services between non-homogeneous hardware components. The most common example of this service is the interchange of information between a computer and a printing device. These services include font information exchange, bar coding, optical disk handling, and graphics device interface (GDI) APIs.

- **Raster/image data interchange** services provide for the handling and manipulation of raster graphics and images. Raster graphics standards are standards for pixel-by-pixel representation of images. Image data standards are standards for the exchange of imagery data, metadata, and attachments to the images.

- **Mapping** services provide formats and facilities for machine-readable mapping, charting, and geospatial data.

- **DoD applications** services are the functional areas unique to DoD missions that are not standardized by nongovernmental standards bodies.

- **Compression** services specify algorithms for compressing data for storage and exchange over a network. Data compression can reduce communications loading by as much as 80 percent without affecting the form of transmitted data. Compression requires application of the same algorithms at the sending and receiving locations. Compression may be used for text and data, still images, and motion images. Compression algorithms for data must be "lossless" so that the expanded output exactly matches the original input. Compression algorithms for still and motion images may be "lossy," where some data may be lost, but the expanded output is not noticeably different from the original input.

2.4.3.5 Graphics Services

Graphics services provide functions required for creating and manipulating pictures. These services include:

- **Raster graphics** represent images as a matrix of dots. Raster graphics images are created by scanners and cameras and are generated by paint software packages. The simplest monochrome bitmap uses one bit (on/off) for each dot. Gray scale bitmaps (monochrome shades) represent each dot with a number large enough to hold all the gray levels. Color bitmaps require sufficient storage to hold the intensity of red, green, and blue, as would a gray scale equivalent.

- **Vector graphics** represent graphical objects as sets of endpoints for lines, curves, and other geometric shapes with data about width, color, and spaces bounded by lines and curves. The entire image commonly is stored in the computer as a list of vectors called a display list. Vector graphics are used when geometric knowledge about the depicted object is needed. Geometric shapes keep their integrity: a line always can be separately selected, extended, or erased. Today, most screens are raster graphics displays (composed of dots), and the vectors are put into the required dot patterns (rasters) by hardware or software. Vector graphics systems must be supplemented by data interchange standards, such as Initial Graphics Exchange Specification (IGES), Computer Graphics Metafile (CGM), and the Standard for the Exchange of Product Model Data (STEP).

- **Device interfaces** provide API services for accessing graphics devices, such as monitors, scanners, printers, etc.

2.4.3.6 Communications Services

Communications services are provided to support distributed applications requiring data access and applications interoperability in heterogeneous or homogeneous networked environments.

- **Application** services are the functions and interfaces that reside on the underlying network and communications system protocol software and are used by applications. These services are based on the presentation and application layers (layers 6 and 7) of the OSI Reference Model.

- **Transport** services perform a variety of functions concerned primarily with the end-to-end transmission of data across a network and end-to-end reliability. The services performed include end-to-end error detection and recovery, regulating flow control, and managing the quality of service. Transport services correspond to the transport and session layers (layers 4 and 5) of the OSI Reference Model.

- **Subnetwork technologies** services support access to LANs and other networks based on the physical, data link, and network layers (layers 1, 2, and 3) of the OSI Reference Model. This area includes LANs, point-to-point communications, packet switching, circuit switching, and military-unique data communications.

2.4.3.7 Operating System Services

Operating system services are the core services needed to operate and administer the application platform and provide an interface between the application software and the platform. Application programmers will use operating system services to access operating system functions. To separate sensitive data within an information system, the kernel must include mechanisms to control access to that information and to the underlying hardware. Security services are defined in Section 2.4.4.2. Operating system services include:

- **Kernel operations** provide low-level services necessary to create and manage processes, execute programs, define and communicate signals, define and process system clock operations, manage files and directories, and control input/output processing to and from peripheral devices. Thread services provide an underlying service used for multiple concurrent executions within a single computer process. They are designed to allow independent operation and are essential for functions such as multiple process communications.

- **Real-time extension** services support event-driven processes supporting management and actuation of physical processes. For this reason, they are often referred to as sensor-based systems. They are designed to handle and process interrupts from a variety of sources (typically involving some kind of sensor device or timer), process associated information through some type of capture or control algorithm, and respond, if necessary, with an appropriate signal to a control or actuation device.

- **Clock/calendar** services provide mechanisms for measuring the passage of time and maintaining the system time. This includes clocks and timers, real time timers, and distributed timing services.

- **Fault management** includes the prevention, isolation, notification, diagnosis, and correction of fault conditions, which arise whenever a malfunction or abnormal behavior results or may result in an error, outage, or degradation of services. Fault management services allow a system to react to the loss or incorrect operation of system components, and they encompass services for fault detection, isolation, diagnosis, recovery, and avoidance.

- **Shell and utilities** include mechanisms for services at the operator level, such as comparing, printing, and displaying file contents; editing files; searching patterns; evaluating expressions; logging messages; moving files between directories; sorting data; executing command scripts; scheduling signal execution processes; and accessing environment information.

- **Operating system object** services define the rules for creating, deleting, and managing objects.

- **Media handling** services provide for disk and tape formatting for data and interchange of data with applications.

2.4.4 Application Platform Cross-Area Services

Besides the service areas delineated by functional category as presented in Section 2.4.3, another category of services and requirements affects the basic information system architectures within the DoD. Treated in a manner similar to those in POSIX.0, these services are referred to as cross-area services and have a direct effect on the operation of one or more of the functional service areas. In some cases, the cross-area services affect each of the functional service areas in a similar fashion, while in other cases, the cross-area service has an influence that is unique to that particular service area. The discussion of the cross-area services is consolidated here in a single location within this document in order to provide a coherent perspective when addressing that service.

The cross-area services presently identified and addressed in this section include internationalization, security, system management, and distributed computing. As the reference model evolves, the cross-area services category will be reexamined for additional components or for reallocation into a functional service area of its own.

2.4.4.1 Internationalization Services

As a practice, information system developers have generally designed and developed systems to satisfy a focused set of requirements that are relevant to a specific market segment. That specific market segment may be a nation or a particular cultural market. To make that information system viable, or marketable, to a different segment of the market, a full re-engineering process was usually required. Users or organizations that needed to operate in a multinational or multicultural environment typically did so with multiple, generally incompatible information processing systems. NATO is an example where a number of countries come together to work toward a common goal yet must deal with a diversity of languages and cultures in their day-to-day operations.

Within the context of the TRM, internationalization provides a set of services and interfaces that allow a user to define, select, and change between different culturally related application environments supported by the particular implementation.

- **Character sets and data representation** services include the capability to input, store, manipulate, retrieve, communicate, and present data independently of the coding scheme used. This includes the capability to maintain and access a central character-set repository of all coded character sets and special graphical symbology used throughout the platform, including the appropriate modifications of GUI screens to match character set conventions. Character sets will be uniquely identified so that the end user or application can select the coded character set to be used. This system-independent representation supports the transfer (or sharing) of the values and syntax, but not the semantics, of data records between communicating systems. The specifications are independent of the internal record and field representations of the communicating systems. Also included is the capability to recognize the coded character set of data entities and subsequently to input, communicate, and present that data.

- **Cultural convention** services provide the capability to store and access rules and conventions for cultural entities maintained in a cultural convention repository. These repositories should be available to all applications and be capable of being sorted based upon local rules defined in the repository.

- **Native language support** services provide the capability to support more than one language simultaneously. Messages, menus, forms, and on-line documentation would be displayed in the language selected by the user. Input from keyboards that have been modified locally to support the local character sets would be correctly interpreted.

- **Related standards and programs** - TBD

2.4.4.2 Security Services

Different groups of individuals within and across the various DoD mission areas need to work with specific sets of data elements. Access to these sets of data elements is to be restricted to authorized users. Satisfaction of this requirement generally has been accomplished by the implementation of separate information systems. Organizations cannot continue to afford to implement separate information systems to satisfy this requirement, nor is it effective to require the user to change interface components every time the need arises to operate with a different restricted data set. Significant benefit will be realized when an individual information system can effectively support the needs of different groups of users and data sets. Such an information system will allow multiple groups to share information systems and data while guaranteeing the separation of data and users as necessary through the use of multi-level security operating systems.

In multi-level security operating systems, the kernel will play the prime role in permitting platforms to handle multiple information domains (security contexts) simultaneously. The separation kernel will be trusted software; that means it will be evaluated in accordance with the requirements stipulated in the documents cited in Volume 7. The separation kernel will mediate all use of the basic information system resources and will provide for strict separation among multiple security contexts by creating separate address spaces for each of them. The separation kernel will provide separation among process spaces by using the protection features of the platform hardware (e.g., processor state registers, memory mapping registers).

The DGSA does not envision security-critical functions being part of these other operating system components. The DGSA envisions such untrusted software performing operations with basic system resources only through invocations of security-critical functions mediated by the separation kernel.

Security services are necessary to protect sensitive information in the information system. The appropriate level of protection is determined based upon the value of the information to the mission-area end users and the perception of threats to it. The information system integrator will need to work with the designated approving authority (DAA) to identify the required level of security protection and acceptable mechanisms for satisfying the requirements. Information system security services are depicted as cross-area services in Figure 2-2 because the mechanisms implemented to provide them may be part of multiple platform service areas. The DGSA currently identifies implementations of security service protection mechanisms in the platform as part of the network and operating system service areas.

The DGSA identifies the following security services that may need to be provided through implementations in information system components. The first five of these services are consistent with the definitions contained in ISO 7498-2, a standard focusing on security related to open systems interconnection communications. The DGSA extends the ISO 7498-2 definitions to apply to more than communications and identifies availability as a security service.

- **Architectures and applications** provide standards, guidance, and frameworks that help to define security architectures and the placement of security into specific applications and are intended to provide guidance to standards developers. They do not provide implementable specifications against which conformance can be claimed.

- **Authentication** service ensure system entities (processes, systems, and personnel) are uniquely identified and authenticated. The granularity of identification must be sufficient to determine the processes, system, and personnel's access rights. The authentication process must provide an acceptable level of assurance of the professed identity of the entities.

- **Access control** service prevents the unauthorized use of information system resources. This service also prevents the use of a resource in an unauthorized way. This service may be applied to various aspects of access to a resource (e.g., access to communications to the resource, the reading, writing, or deletion of an information/data resource, the execution of a processing resource) or to all accesses to a resource. Security labels are used to manage access and privileges, which are managed for all entities, whether individual users, groups of users, resources, or processes.

- **Integrity** service ensures protection of the system through open system integrity, network integrity, and data integrity. This ensures that data is not altered or destroyed in an unauthorized manner. This service applies to data in permanent data stores and to data in communications messages.

- **Confidentiality** service ensures that data is not made available or disclosed to unauthorized individuals or computer processes through the use of data encryption, security association, and key management. This service will be applied to devices that permit human interaction with the information system. In addition, this service will ensure that observation of usage patterns of communications resources will not be possible.

- **Non-repudiation** services include open systems non-repudiation, electronic signature, and electronic hashing. Non-repudiation services ensure that senders and recipients cannot deny the origin or delivery of data. Non-repudiation mechanisms can be used to validate the source of software packages or to verify that hardware is unchanged from its manufactured state.

- **Availability** service ensures that timely and regular communications services are available. These services are intended to minimize delay or non-delivery of data passed on communications networks. These services include protecting communications networks from accidental or intentional damage and ensuring graceful degradation in communications service.

- **System management** services encompass those security functions required to maintain an operationally secure system. These services include analysis areas such as certification and accreditation and risk management, as well as operationally motivated concerns such as alarm reporting, audit, and cryptographic key management.

- **Security labeling** is the data bound to a resource (which may be a data unit) that names or designates the security attributes of that resource. Security labeling includes security labeling for the following major service areas: user interface, data management, data interchange, graphics, network (data communications), system, and distributed computing.

- **Information system security management** services are concerned with the installation, maintenance, and enforcement of information domain and information system security policy rules in the information system intended to provide these security services. In addition to these core services, security management requires event handling, auditing, and recovery. Standardization of security management functions, data structures, and protocols will enable interoperation of security management application programs (SMAPs) across many platforms in support of distributed security management. Areas for security management standardization are described in Volume 6.

Classes of managed objects for security management are security policies, security services, and security mechanisms. Some information is managed for specific information domains and for the platform in a distributed or non-distributed environment. The items of information that might be included in the security management information base (SMIB) for each information domain and for the platform itself are described in Volume 6.

2.4.4.3 System Management Services

Information systems are composed of a wide variety of diverse resources that must be managed effectively to achieve the goals of an open system environment. While the individual resources (such as printers, software, users, processors) may differ widely, the abstraction of these resources as managed objects allows for treatment in a uniform manner. The basic concepts of management, including operation, administration, and maintenance, may then be applied to the full suite of OSE components along with their attendant services.

Work on systems management services and attendant standards is ongoing. This work is based predominantly on the Open System Interconnection (OSI) network management framework, which applies mainly to networks and the individual nodes on the networks. There is, however, an overlap among certain types of network management functions and individual system management functions. This overlapping area applies equally to networks and individual systems and forms the basis for the OSI approach to systems and network management. Other system management functions in the typical operating system sense are also being addressed and need to be integrated into the overall systems and network management framework. Systems management functionality may be divided according to the management elements that generically apply to all functional resources, which are state management, configuration control, performance management, fault management, user/group management, usage management and other management.

This breakout of system management services parallels the breakout of OSI network management, thereby presenting an overall coherent framework that applies equally to networks and the individual nodes of the networks. Many of the specific services have no formal standards work in progress; however, industry consortia and others are addressing selected areas.

One important consideration of the standards supporting the services in this area is that they should not enforce specific management policies but rather enable a wide variety of different management policies to be implemented, selected according to the particular needs of the end-user installations.

- **State management** services provide for mechanisms that monitor, maintain, and change the state of the system or components of the system.

- **Configuration control** services address four basic functions: identification, control, status accounting, and verification. Identification involves identifying and specifying all component resources. Control implies the ability to freeze configuration items and then to change them only through a process involving agreement of appropriate name authorities. Status accounting involves the recording and report of all current and historical data about each configuration item. Verification consists of a series of reviews and audits to ensure conformity between the actual configuration item and the information recorded about it. The services which provide these functions include software distribution and license management.

- **Performance management** services allow information technology resources to be managed efficiently. Performance aspects of hardware, software, and network components must be monitored and subsequently made available to the system manager. The manager must then have access to services and parameters with which to tune the system to meet performance targets. This is accomplished through batch scheduling, system resource management, print and storage device management, system startup and shutdown, subsystem management, and communication of management information.

- **Fault management** services allow a system to react to the loss or incorrect operation of system components at various levels (hardware, software, etc.). Fault management involves event management and network error recovery.

- **User/group management** services provide traditional system administration interfaces for administering users and groups. These services are mechanisms for system and network administrators to use when implementing a management policy across a system. Administrators can use the services to establish domains and policies for management throughout the system. They can provide the ability for applications to access group and user databases. Users can set up their own areas of management and policies or use system defaults that are included in management services.

- **Usage management and cost allocation** services include the management of software licensing, system cost management, and system resource allocation. Software license management for a system provides license administration, management, and enforcement services that allow more detailed, firm, and equitable licensing terms for users, and better protection against illegal software usage for vendors. Cost management services provide the ability to cost services for charging and reimbursement and to measure and prioritize resource usage.

System resource allocation allows system administrators to control the amount of system resources available to users.

- **Other management** services include the following services which do not fit cleanly into any other management area: database administration, object-oriented database management, floppy disk formatting and handling, POSIX tape labeling and tape volume processing, and print management. Database and object-oriented database administration provide facilities and interfaces for the management of databases and object-oriented databases, respectively. Floppy disk formatting and handling standards provide formats and interfaces for the exchange, backup, and restoration of data to or from floppy disks. POSIX tape labeling and tape volume processing provide for standardized methods of handling and reading data stored on tape media and containing certain types of administrative information automatically readable by tape-handling software. Print management services are used by management and user applications to send a file to a printer, cancel a print job, and get printer status information. (Security system management services are discussed above, in Section 2.4.4.2, as part of security services.)

System management application processes, using information in the information base, will be used to establish the required security contexts for interactive communications among distributed platforms operating in various information domains simultaneously. This approach is intended to support secure distributed computing services. System management application processes will also be used to provide the security protection of store-and-forward communications in which the requisite security contexts cannot be handled within the message.

2.4.4.4 Distributed Computing Services

Distributed computing services provide specialized support for applications that may be physically or logically dispersed among computer systems in a network yet wish to maintain a cooperative processing environment. The classical definition of a computer becomes blurred as the processes that contribute to information processing become distributed across a facility or a network. As with other cross-cutting services, the requisite components of distributed computing services typically exist within particular service areas. They are described below to offer a coherent view of this important service.

- **Client/server** services provide support for computing services which are partitioned into requesting processes (clients) and providing processes (servers), whether on the same platform or in a distributed environment.

- **Object** services support the definition, instantiation, and interaction of objects in a distributed environment, and include services which handle operating system bindings, message transport and delivery, and data persistence.

- **Remote access** services provide location transparency functionality for distributed computing services, allowing users and client processes to access appropriate systems resources (files, data, processes) without regard to the location of either.

This page intentionally left blank.

APPENDIX A

REFERENCES

Note: References appearing in this section represent documents used in preparation of the TAFIM, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

1. Executive Level Group for Defense Corporate Information Management, A Plan for Corporate Information Management for the Department of Defense, 11 September 1991.

2. Application Portability Profile (APP), The U.S. Government's Open System Environment Profile OSE/1 Version 2.0, NIST SP-500-210, June 1993.

3. Army Tactical Command and Control Information System, Technical Standards for CCISs, Third Edition, 21 January 1992.

4. AT&T, Open Look Graphical User Interface Trademark Guide, 1990.

5. DEPSECDEF Memo, 14 January 91, Implementation Plan for Corporate Information Management, with Enclosure 774.

6. DIA, DIA Information System Architecture Standards and Products, 10 May 1990.

7. DLA Office of Information Systems and Technology, Information Resources Management Environment Vision and Prescription, Version 1.1, April 1991.

8. DoD Intelligence Information System (DODIIS) Reference Model for the 1990s, Defense Intelligence Agency, Draft, 14 May 91.

9. IEEE Draft Guide to the POSIX Open System Environment (P1003.0/D15), Institute of Electrical and Electronics Engineers, Inc., May 1992.

10. NIST, FIPS 146-2, Profiles for Open Systems Networking Technologies, 1996.

11. NIST Special Report 500-187, Application Portability Profile (APP): The U.S. Government's Open System Environment Profile OSE/1, Version 1.0, May 1991.

12. NIST Special Publication 500-163, Government Open Systems Interconnection Profile (GOSIP) User's Guide, 2nd Edition.

13. NIST Special Publication 500-201, Reference Model for Frameworks of Software Engineering Environments (Technical Report ECMA TR/55, 2nd Edition), December 1991.

14. OSF, OSF/Motif Application Environment Specification User Environment, Volume 1.0, Rev A, 1990.

15. OSF, OSF/Motif Programmer's Guide, Rev 1.0, 1990.

16. OSF, OSF/Motif Style Guide, Rev 1.0, 1990.

17. OSF, OSF/Motif User's Guide, Rev 1.0, 1990.

18. Plan for Implementation of Corporate Information Management in DoD, ASD/C3I, 8 January 1991.

19. SECDEF Memo, November 16, 1990, Implementation of Corporate Information Management Principles w/Enclosure.

20. SM-684-88, Policy and Procedures for Management of Command, Control, and Communications Systems, JCS, undated.

21. Sun Microsystems, Inc., Open Look Graphical User Interface Application Style Guidelines, 1989.

22. Sun Microsystems, Inc., Open Look Graphical User Interface Functional Specifications, 1989.

23. Strategies for Open Systems, Stage Two: The Experience With Open Systems, DMR Group, Inc., Boston, 1990, pp. 196.

24. X/OPEN Company, Ltd., X/OPEN Portability Guide, Version 3 (XPG3), 1988.

25. X/Open Portability Guide, Issue 3, Volumes 1-7, X/Open Company, Ltd., Prentice Hall, Inglewood Cliffs, NJ, 1988.

APPENDIX B

ACRONYMS

AITS	Adopted Information Technology Standards
AMWG	Architecture Methodology Working Group
ANSI	American National Standards Institute
API	Application Program Interface
APP	Application Portability Profile
ASC	Accredited Standards Committee
ASD(C3I)	Assistant Secretary of Defense, Command, Control, Communications, and Intelligence
BSA	Base Service Area
CAD	Computer-Aided Design
CALS	Computer-Aided Acquisition and Logistic Support
CAP	Communication-Electronics Accommodation Program
CASE	Computer-Aided Software Engineering (See ISEE)
CGM	Computer Graphics Metafile
CIM	Corporate Information Management
COTS	Commercial-Off-the-Shelf
DAA	Designated Approving Authority
DBMS	Database Management System
DDI	Director of Defense Information
DEPSECDEF	Deputy Secretary of Defense
DGSA	Defense Goal Security Architecture
DISA	Defense Information Systems Agency
DLA	Defense Logistics Agency
DoD	Department of Defense
DODIIS	DoD Intelligence Information System
DSRS	DoD Software Reuse System
EEI	External Environment Interface
FIPS	Federal Information Processing Standard
GOTS	Government-Off-the-Shelf

GUI	Graphical User Interface
HCI	Human Computer Interface
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IGES	Initial Graphics Exchange Specification
INX	Information Exchange
ISO	International Organization for Standardization
IT	Information Technology
ITPB	Information Technology Policy Board
ITSI BBS	Information Technology Standards Information Bulletin Board System
JIEO	Joint Interoperability and Engineering Organization
JTC	Joint Technical Committee
MSA	Major Service Area
MLSA	Mid-Level Service Area
NATO	North Atlantic Treaty Organization
NDI	Nondevelopmental Item
NIST	National Institute of Standards and Technology
OA&M	Operation, Administration, and Maintenance
ODT	Optical Digital Technologies
OSD	Office of the Secretary of Defense
OSE	Open System Environment
OSF	Open Software Foundation
OSI	Open System Interconnection
POSIX	Portable Operating System Interface (for Computer Environments)
SECDEF	Secretary of Defense
SMAP	Security Management Application Program
SMIB	Security Management Information Base
STEP	Standard for the Exchange of Product Model Data
TRM	Technical Reference Model
UI	UNIX International

APPENDIX C

OFFICE OF THE ASSISTANT SECRETARY OF DEFENSE MEMORANDA CONCERNING OPEN SYSTEMS IMPLEMENTATION AND THE TECHNICAL REFERENCE MODEL

This appendix provides the text of a memorandum from the Assistant Secretary of Defense concerning open systems implementation and the Technical Reference Model, dated 30 March 1995.

MEMORANDUM FROM
THE ASSISTANT SECRETARY OF DEFENSE

March 30, 1995

MEMORANDUM FOR UNDER SECRETARIES OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY (RD&A)
ASSISTANT SECRETARY OF THE NAVY (RD&A)
ASSISTANT SECRETARY OF THE AIR FORCE
(ACQUISITION) (SAF/AQ)
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM), Version 2.0

My memorandum dated June 23, 1994 established the TAFIM as the single framework to promote the integration of Department of Defense (DoD) information systems, expanding the opportunities for interoperability and enhancing our capability to manage information resources across the Department. The latest version of the TAFIM, Version 2.0, is complete and fully coordinated. Version 2.0 consists of seven volumes as shown in the attachment. The TAFIM will continue to guide and enhance the evolution of the Department's information systems technical architectures.

I want to reiterate two important points that I made in my June 1994 memorandum. First, the Department remains committed to a long range goal of an open systems environment where interoperability and cross functional integration of our systems and portability/reusability of our software are key benefits. Second, the further selection and evaluation of migration systems should take into account this long range goal by striving for conformance to the TAFIM to the extent possible.

Effectively immediately, new DoD information systems development and modernization programs will conform to the TAFIM. Evolutionary changes to migration systems will be governed by conformance to the TAFIM.

The TAFIM is maintained by the Defense Information Systems Agency (DISA) and is available electronically via the DISA On-Line Standards Library. Hardcopy is available through the Defense Technical Information Center. The TAFIM is an evolving set of documents and comments for improving may be provided to DISA at any time. The DISA action officer is Mr. Bobby Zoll, (703) 735-3552. The OSD action officer is Mr. Terry Hagle, (703) 604-1486.

s/Emmett Paige, Jr.

APPENDIX D

PROPOSING CHANGES TO TAFIM VOLUMES

D.1 INTRODUCTION

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section D.2. Guidance on the use of the format is provided in Section D.3.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown in Section D.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail: **TAFIM**
Booz, Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/824-3770**; indicate "TAFIM" on cover sheet.

D.2 TAFIM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

D.3 FORMAT GUIDANCE

The format in Section D.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the

Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part **(D.2 a)** of the format would identify that individual. The information in the **Document Identification** part of the format **(D.2 b)** is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts **(D.2 c, D.2 d, or D.2 n)** of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: 'The working baseline will only be available to the TAFIM project staff.'" The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The D.2 c (5), D.2 d (5), or D.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in D.2 c (6), D.2 d (6), or D.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

This page intentionally left blank.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 3: Architecture Concepts and Design Guidance

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents

- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.

- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 3 of the TAFIM, contains minor substantive changes from Volume 3 of Version 2.0, most of which are intended to resolve internal inconsistencies or to bring the guidance provided in this volume more in line with current policies. Work remains to be done to fully reflect the impact of the documents and decisions noted above; this edition of the TAFIM has been released to serve as a baseline and to make available throughout the DoD community the additions and modifications that have been implemented to date.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version

3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.
- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.
- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. The Defense Information Systems Agency (DISA) is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The **latest authorized versions** of the TAFIM volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

CONTENTS

FIGURES

1.0 INTRODUCTION

The Technical Architecture Framework for Information Management (TAFIM) characterizes an information system as composed of data, mission-specific applications, and a technical infrastructure consisting of support applications, application platforms, and communications networks. This document presents technical architecture concepts and design guidance for information systems in the Department of Defense (DoD). As part of the TAFIM, this volume provides guidance for the evolution of the DoD's technical infrastructure in support of specific mission requirements. The data and mission-specific software architectures are critical elements in information system development. Guidance on their development and use will be provided in separate documents outside of the TAFIM.

1.1 SCOPE

Volume 3 provides concepts and design guidance that will help architects, integrators, and system designers to develop information systems technical architectures. These concepts and guidance should be considered in the context of the Technical Reference Model presented in Volume 2.

The contents of this volume contrast with the TAFIM Volume 2, which describes services and interfaces between entities. This volume addresses components and the allocation of services to the components.

1.2 DOCUMENT ORGANIZATION

Volume 3 of the TAFIM consists of three sections and four appendices. Section 2 discusses several architecture concepts of interest to architects, designers, and developers. Section 3 presents design guidance based on availability and maturity of technology. Appendix A provides acronyms and definitions. Appendix B provides a definition and discussion of open systems. Appendix C contains references. Appendix D contains a template for submitting comments on this volume.

This page intentionally left blank.

2.0 ARCHITECTURE CONCEPTS

The DoD vision described in Volume 1 depicts a future information technology environment that includes:

- Shared global databases
- Shared utility services
- Centrally managed and operated backbone network
- Distributed data
- Distributed processes
- Standard user interface
- Transparent information, computing, and information utility
- Individual tailoring of information resources.

The new DoD architectural framework supports an orderly migration from existing legacy systems to DoD standard systems operating in a distributed computing environment that incorporates the above features. Over time, systems will be reengineered or developed to conform to the architecture concepts and standards in the TAFIM. As this occurs, a distributed computing environment will evolve, where processing nodes are constructed to provide services to meet the requirements of the DoD community.

Figure 2-1 depicts a distributed computing environment, which includes platforms and, optionally, support applications and mission-area applications. The external environment shown in the figure consists of entities that are external to the application software and the platform (e.g., users, communications networks). The actual features of an implementation will be dictated by functional requirements and processing efficiency. The enterprise backbone network in this distributed computing environment will provide end-to-end communications services that connect all of the processing nodes, down to an individual's workstation. Through the network, authorized users and applications will have access to all required data and processing resources without having to know the location of the resources. This will include access to shared enterprise global databases and utility services by distributed applications and fixed and mobile users. Resources will be provided through a transparent combination of local and remote processes. Distribution of redundant enterprise data and processing resources across multiple processing nodes will provide processing efficiency, reliability, and survivability.

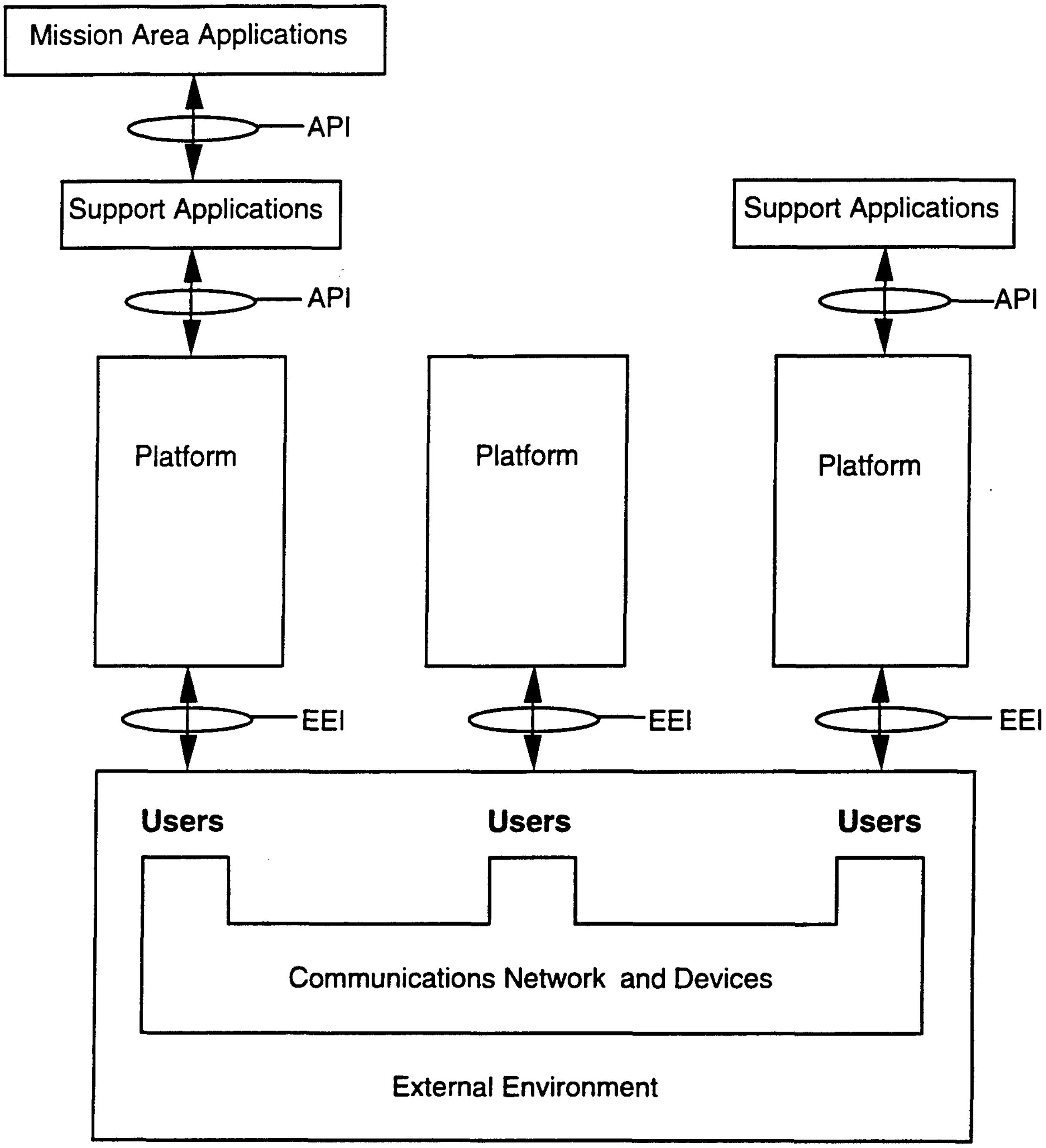


Figure 2-1. Distributed Computing Context

2.1 ARCHITECTURE DEFINITIONS

This section sets the stage for the architecture concepts to be described. It provides basic definitions within the context of a model for architectures.

2.1.1 Architecture Model

A technical architecture defines components, interfaces, services, and the framework within which they interoperate. Components provide either information processing or communications services. A component provides a complete service or part of a service. A component may also provide more than one service. Interfaces link components so that they may interoperate. Figure 2-2 depicts a model of these relationships.

Figure 2-3 depicts service components and their interfaces. The TAFIM provides guidance on the following interfaces: a) between applications (mission-area and support applications) and service components, b) between separate service components, and c) between service components and the external environment.

Services are invoked through an interface, which defines the access rules. Two types of interfaces are described in the Technical Reference Model: an application program interface (API), which defines the rules and protocols used by an application to invoke a service; and an external environment interface (EEI), which defines the rules and protocols for invoking the external environment services. EEI services are provided to support users, peripherals, and remote processors. Volume 2 defines an API as the interface that enables applications to invoke application platform services. To satisfy DoD Information Management (IM) requirements, the TAFIM has applied the definition of an API to any service provided to an application through a programming interface. This interpretation was necessary to meet two distinct requirements.

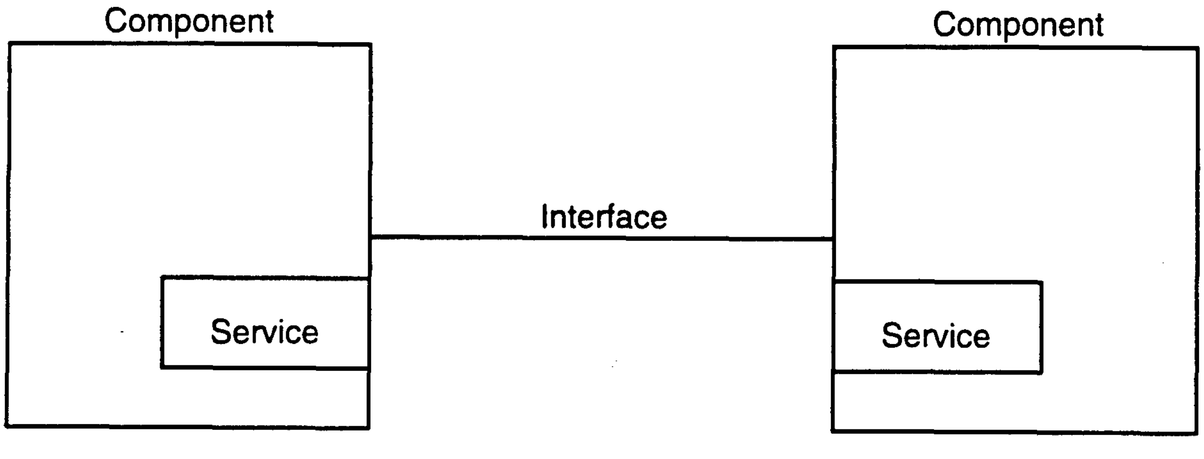


Figure 2-2. Model of Information System Architecture

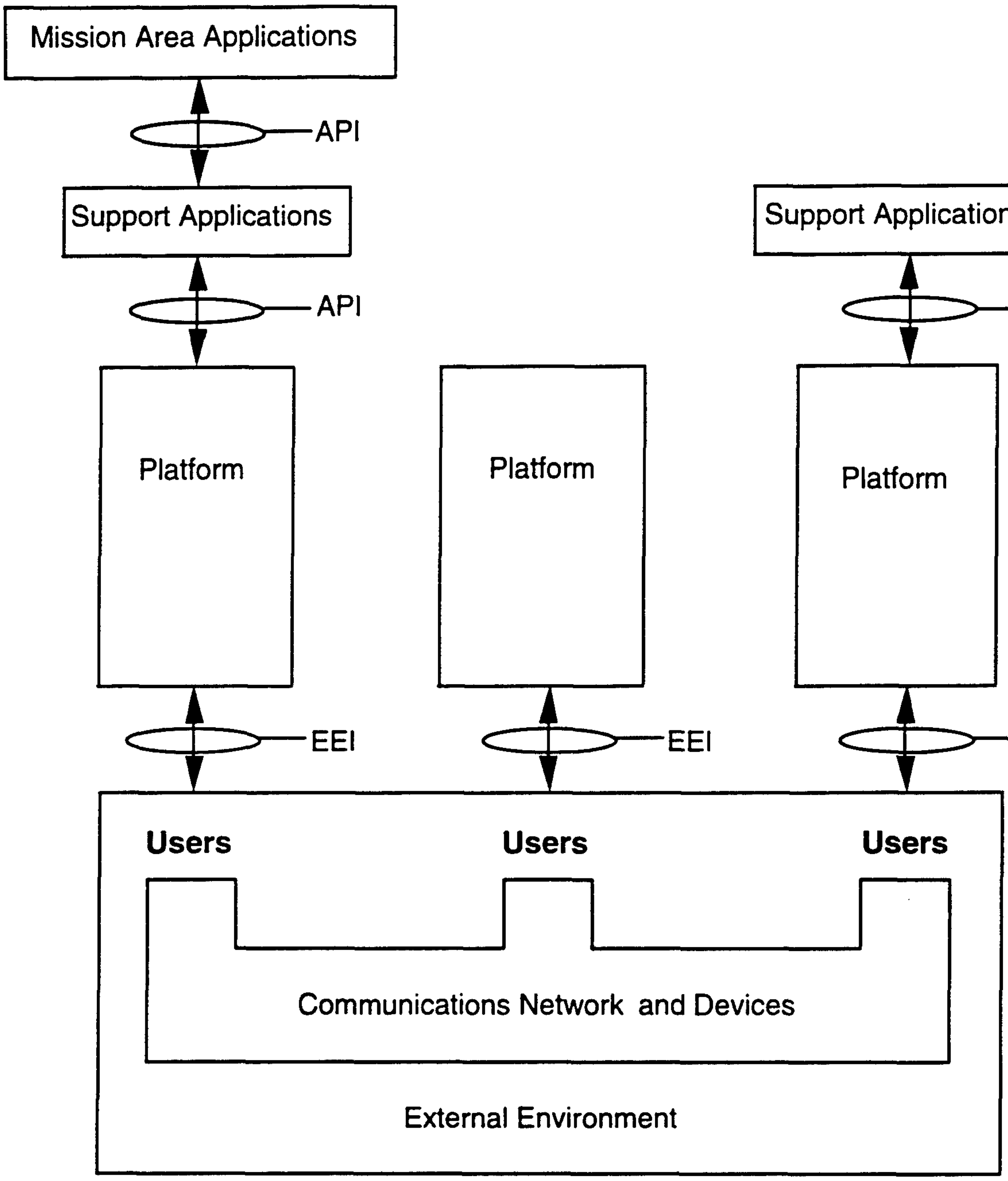


Figure 2-1. Distributed Computing Context

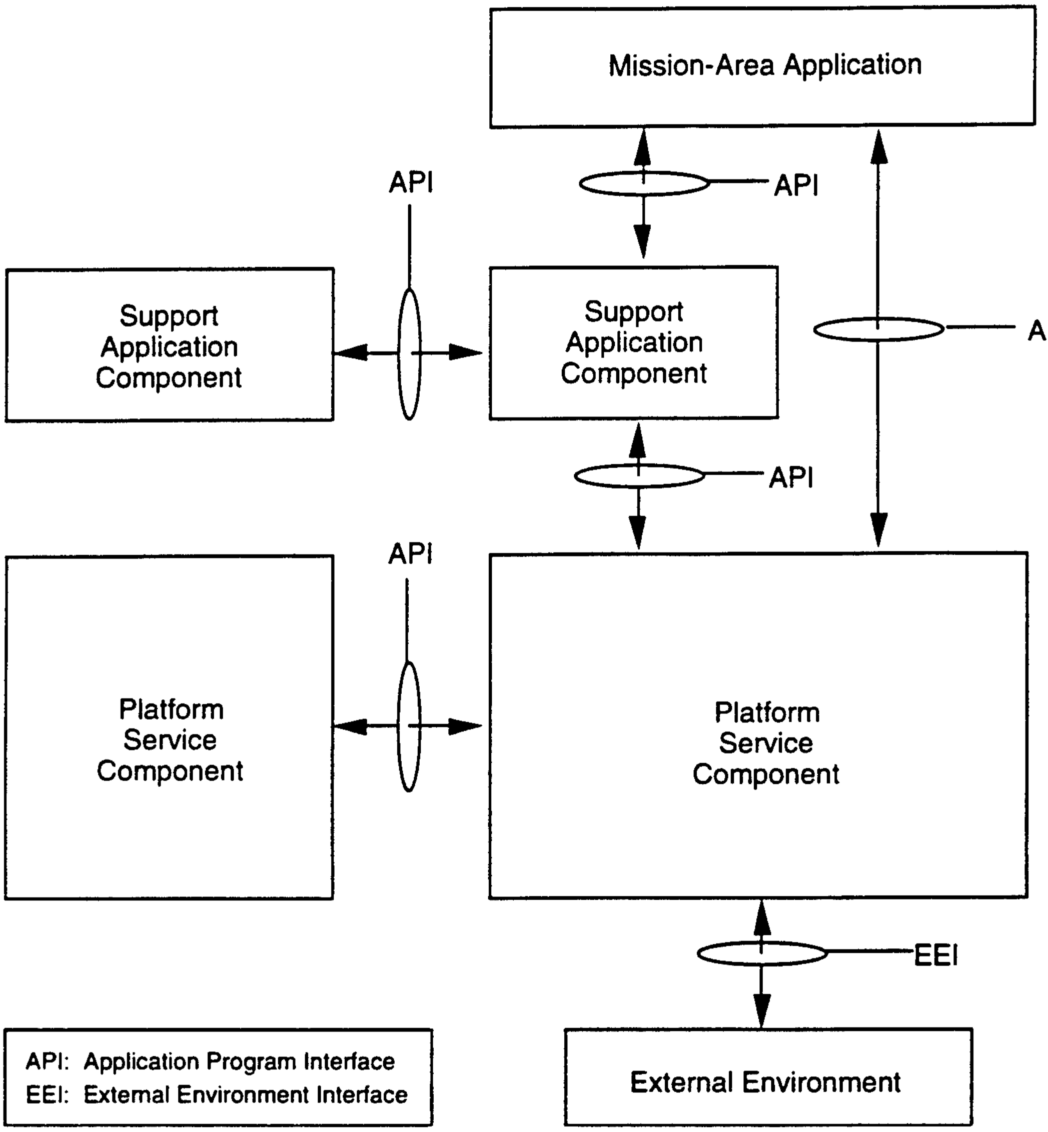

Mission-Area Application
API
API
Support Application Component
Support Application Component
API
API
API
Platform Service Component
Platform Service Component
EEI
API: Application Program Interface
EEI: External Environment Interface
External Environment

Figure 2-3. Technical Architecture Service Context

First, it supports the use of services not provided by the application platform. Recognizing that many reusable services are not covered under the platform service categories, the TAFIM has split the applications into mission-area applications and support applications. The support applications provide common reusable services, such as word processing and electronic mail (E-mail), to mission-area applications and other support applications. To support mission-area and support application portability, DoD has a requirement for standard application interfaces to these services. Applying the definition of the API to address this interface supports the potential future migration of services between the support applications and platforms.

Second, this expansion supports the distribution of computing and communications resources throughout the network. The use of one platform component's service by another platform component is defined as a system internal interface (SII) by POSIX P1003.0. The DoD requirement is for platform components to use the same API that an application uses when requesting services from other platform components. For example, if an Information Resource Dictionary System (IRDS)-compliant dictionary has a requirement to store data in a database, it should use the Structured Query Language (SQL) API to invoke the services of a Relational Database Management System (RDBMS). This will minimize unique dependencies between platform components, enhancing the capability to replace one platform component with another. It will also provide DoD with the maximum flexibility possible in distributing computing and communication resources throughout the network.

2.1.2 Architecture Views

Depending on the area of responsibility of the architect or designer, an architecture may be viewed from different perspectives. For example, the designer responsible for computing perceives the architecture with a different focus than the designer responsible for data management. The architect responsible for the overall system has yet another focus. The views presented in the remaining subsections of Section 2 (2.2, 2.3, 2.4, 2.5) describe architecture concepts from different perspectives. Each of these views addresses components, interfaces, and allocation of services critical to the view.

2.2 COMPUTING VIEW

This view of the technical architecture focuses on computing models that are appropriate for a distributed computing environment. To support the migration of legacy systems, the section also presents models that are appropriate for a centralized environment. The definitions of many of the computing models (e.g., host-based, master-slave, and three-tiered) historically preceded the definition of the client/server model, which attempts to be a general-purpose model. In most cases the models have not been redefined in the computing literature in terms of contrasts with the client/server model. Therefore, some of the distinctions of features are not always clean. In general, however, the models are distinguished by the allocation of functions for an information system application to various components (e.g., terminals, computer platforms). These functions that make up an information system application are presentation, application function, and data management.

2.2.1 Client/Server Model

Client/server processing is a special type of distributed computing termed cooperative processing because the clients and servers cooperate in the processing of a total application (presentation, functional processing, data management). In the model, clients are processes that request services, and servers are processes that provide services. Clients and servers can be located on the same processor, different multiprocessor nodes, or on separate processors at remote locations. The client typically initiates communications with the server. The server typically does not initiate a request with a client. A server may support many clients and may act as a client to another server. Figure 2-4 depicts the basic client/server model.

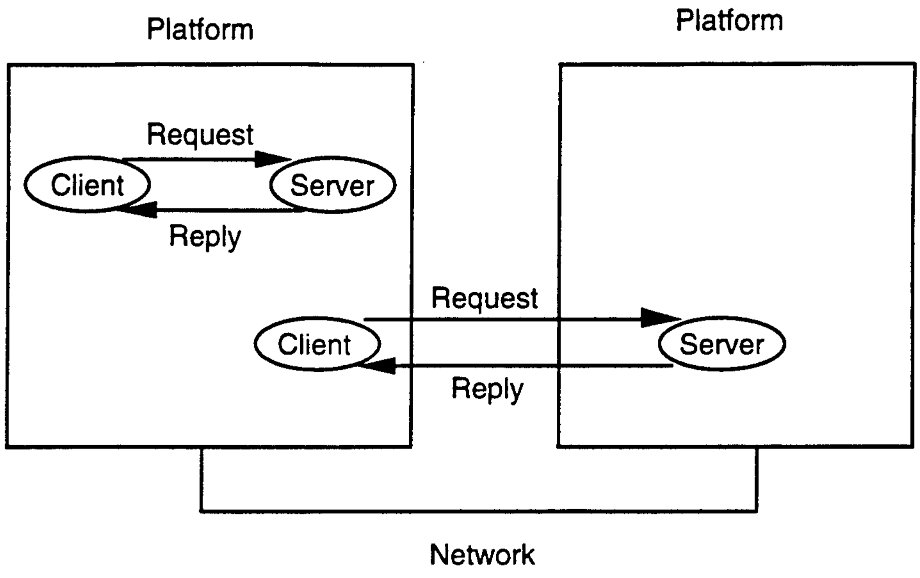


Figure 2-4. Basic Client/Server Model

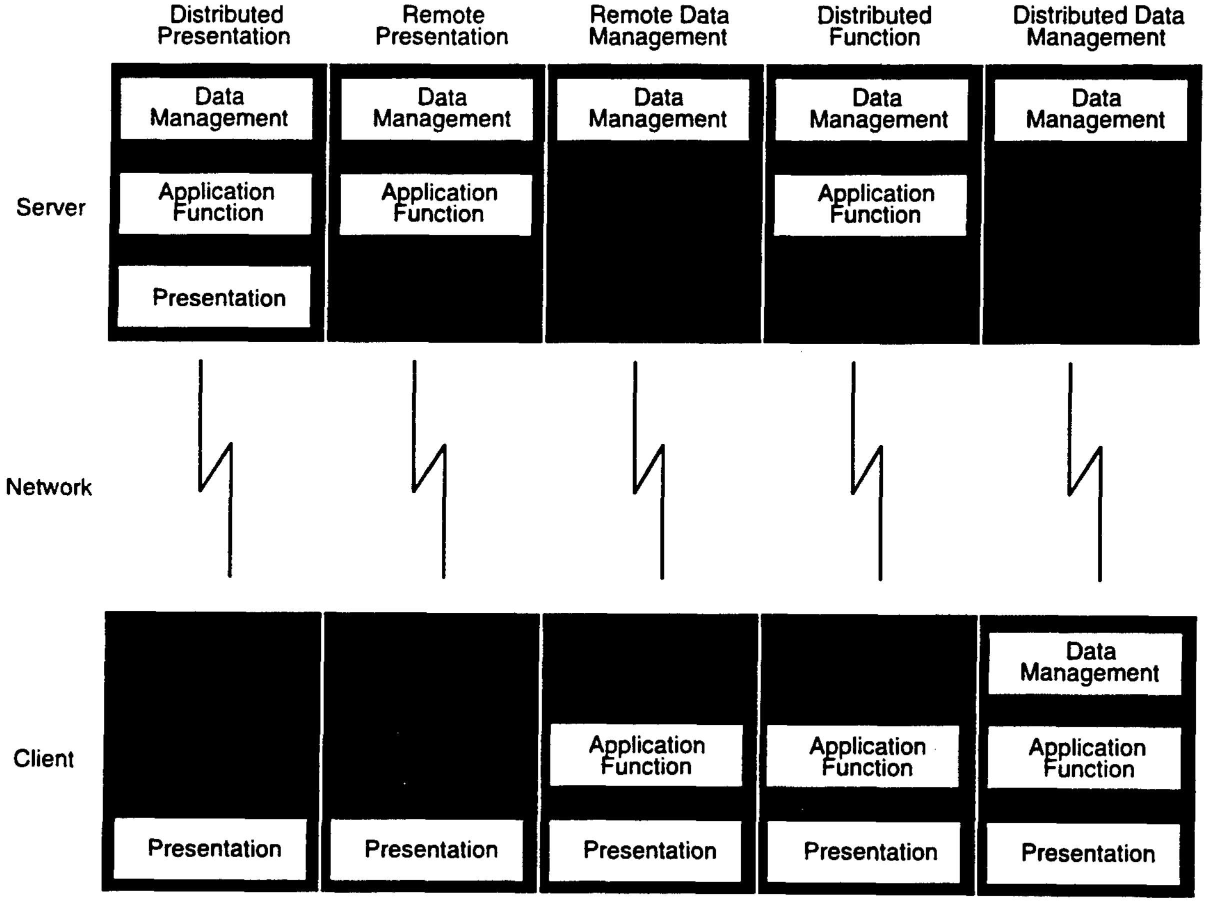
Clients tend to be generalized and can run on one of many nodes. Servers tend to be specialized and run on a few nodes. Clients are typically implemented as a call to a routine. Servers are typically implemented as a continuous process waiting for service requests (from clients). Many client/server implementations involve remote communications across a network. However, nothing in the client/server model dictates remote communications, and the physical location of clients is usually transparent to the server. The communication between a client and a server may involve a local communication between two independent processes on the same machine.

An application program can be considered to consist of three parts–the application function, the presentation, and the data management. In general, any of these can be assigned to either a client or a server. The assignment of each of these program parts to clients and servers can define client/server configurations. The following are five client/server configurations, which demonstrate the flexibility of the client/server model in implementing distributed paradigms. The terms "remote" and "distributed" are from the perspective of the application function portion of the processing:

- Distributed Presentation
- Remote Presentation
- Remote Data Management
- Distributed Function
- Distributed Data Management.

These five client/server configurations, which are based on a Gartner Group Report, are frequently cited in computing literature, but some sources find that they do not represent the current state of commercial-off-the-shelf (COTS) offerings. For example, the Defense Information Systems Agency's (DISA) *Client Server Migration Guidance* presents three models as alternatives to the above popular configurations. They are Presentation Logic Functions, Business Logic Functions, and Data Management Functions. Their rationale is as follows. The distinctions between remote and distributed presentation, data management, and distributed function do not relate easily to COTS products. As an example, the remote data management model states that the presentation functions reside entirely in a client. In practice, virtually every implementation places some functions, however small, in a server. This places these implementations into the distributed presentation model category. Similar situations occur for the other models.

Another client/server configuration growing in popularity is the multitiered architecture. The multitiered architecture and the five popular client/server configurations that are listed above are discussed in the following sections. The five client/server configurations are presented in Figure 2-5.



Source: Gartner Group

Figure 2-5. Client/Server Configurations

2.2.1.1 Distributed Presentation

This model distributes responsibility for presentation between the client and the server. An architecture that implements this model is the X Window architecture. In such an implementation, X terminals are used as client platforms, which contain presentation functions. All other functions (application function and data management), including additional presentation capability, are on the server.

2.2.1.2 Remote Presentation

This configuration separates presentation logic from functional logic by locating the entire presentation function on the client workstation. The client is responsible for the user interface, accepting and validating user input, and sending requests to and receiving results of requests from the server. The advantage of this model is that client and servers are separated by a network, keeping presentation logic off the network. In this scenario, clients request data management and application functional processing services from servers.

2.2.1.3 Remote Data Management

In this configuration, a central server specializes in data management, which might include data security, integrity, and processing database requests from the client. The management of data is separate from the application. This model can be used to support central subject area databases serving one or more remote clients. An example configuration is a database machine or database server attached to clients on workstations.

2.2.1.4 Distributed Function

In this configuration, multiple servers provide specific application processing functions for client applications. The advantages offered through the distribution of application functions include the reduction of redundant code, centralized management and operation of complex processing functions, the ability to distribute some application functions closer to the end user, and the ability to configure and tune specialized servers for maximum processing efficiency. Examples of servers that provide distributed application functions include mail servers, print servers, transaction processors, communication servers, mission-area application servers, and directory servers.

2.2.1.5 Distributed Data Management

In this configuration, the responsibility for data management is split among more than one server. When one data management server, which may or may not be local to the client application, cannot satisfy a request for data, it in turn becomes a client to another data management server that is capable of satisfying the original request. The original client application is unaware that more than one server participated in processing the request. This variation can be introduced during application consolidation and migration, where data is distributed across multiple legacy databases. It can also be used to support an environment where a logical subject area database is spread over several physical databases. An example configuration is a distributed database on more than one platform.

2.2.1.6 The Multitiered Architecture

All of the client/server configurations presented so far in this section (2.2.1) show functions (presentation, application logic, and data management) distributed over two virtual platforms. These can be considered two-tiered architectures. Multitiered client/server architectures with three or more tiers have been proposed and are gaining in popularity.

In multitiered architectures, functions are distributed over multiple virtual (or logical) platforms. These architectures accommodate the partitioning of applications so that user interfaces reside on the user's platform, functional services reside on one or more other networked platforms, and data and legacy systems reside on additional networked platforms.

2.2.2 Host-Based Model

The host-based model is an approach that provides centralized processing on a host machine – that is, it provides no distributed processing. The typical configuration is a mainframe with attached dumb terminals. The central computer does all of the processing (e.g., presentation, application functional processing, data management). Figure 2-6 presents an example host-based configuration.

2.2.3 Master-Slave and Hierarchic Models

In this model, slave computers are attached to a master computer. In terms of distribution, the master-slave model is one step up from the host-based model. Distribution is provided in one direction–from the master to the slaves. The slave computers perform application processing only when directed to by the master computer. In addition, slave processors can perform limited local processing, such as editing, function key processing, and field validation. A typical configuration might be a mainframe as the master with personal computers (PC) as the slaves acting as intelligent terminals, as illustrated in Figure 2-6.

The hierarchic model is an extension of the master-slave model with more distribution capabilities. In this approach, the top layer is usually a powerful mainframe, which acts as a server to the second tier. The second layer consists of local area network (LAN) servers and clients to the first layer as well as servers to the third layer. The third layer consists of PCs and workstations. This model has been described as adding true distributed processing to the master-slave model. Figure 2-6 shows an example hierarchic model in the third configuration.

2.2.4 Peer-to-Peer Model

In the peer-to-peer model there are coordinating processes. All of the computers are servers in that they can receive requests for services and respond to them; and all of the computers are clients in that they can send requests for services to other computers. In current implementations, there often are redundant functions on the participating platforms.

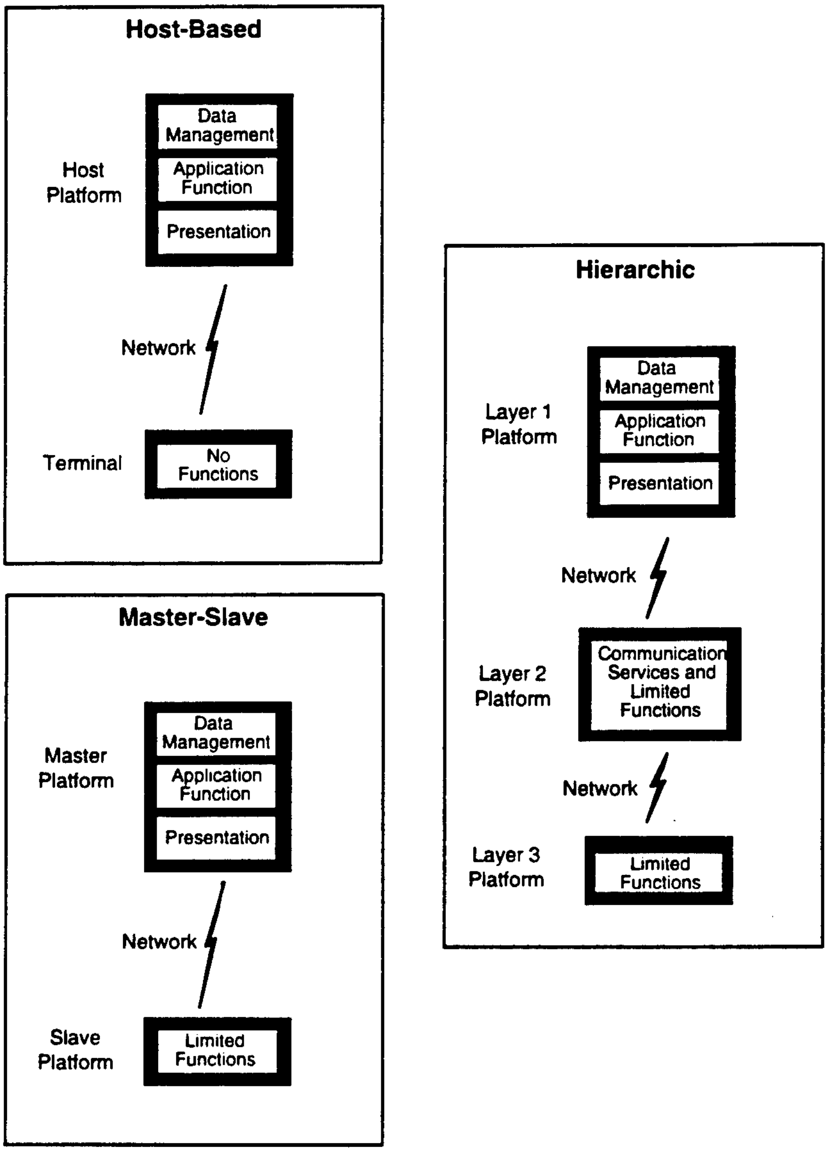


Figure 2-6. Host-Based, Master-Slave, and Hierarchic Models

Attempts have been made to implement the model for distributed heterogeneous (or federated) database systems. This model could be considered a special case of the client/server model, in which all platforms are both servers and clients. Figure 2-7 (A) shows an example peer-to-peer configuration in which all platforms have complete functions.

2.2.5 Distributed Object Management Model

In this model the remote procedure calls typically used for communication in the client/server and other distributed processing models are replaced by messages sent to objects. The services provided by systems on a network are treated as objects. A requester need not know the details of how the object is configured. The approach requires: 1) a mechanism to dispatch messages; 2) a mechanism to coordinate delivery of messages; and 3) applications and services that support a messaging interface. This approach does not contrast with client/server or peer-to-peer models but specifies a consistent interface for communicating between cooperating platforms. It is considered by some as an implementation approach for client/server and peer-to-peer models. Figure 2-7 presents two distributed object model examples. Example B shows how a client/server configuration would be altered to accommodate the distributed object management model. Example C shows how a peer-to-peer model would be altered to accomplish distributed object management.

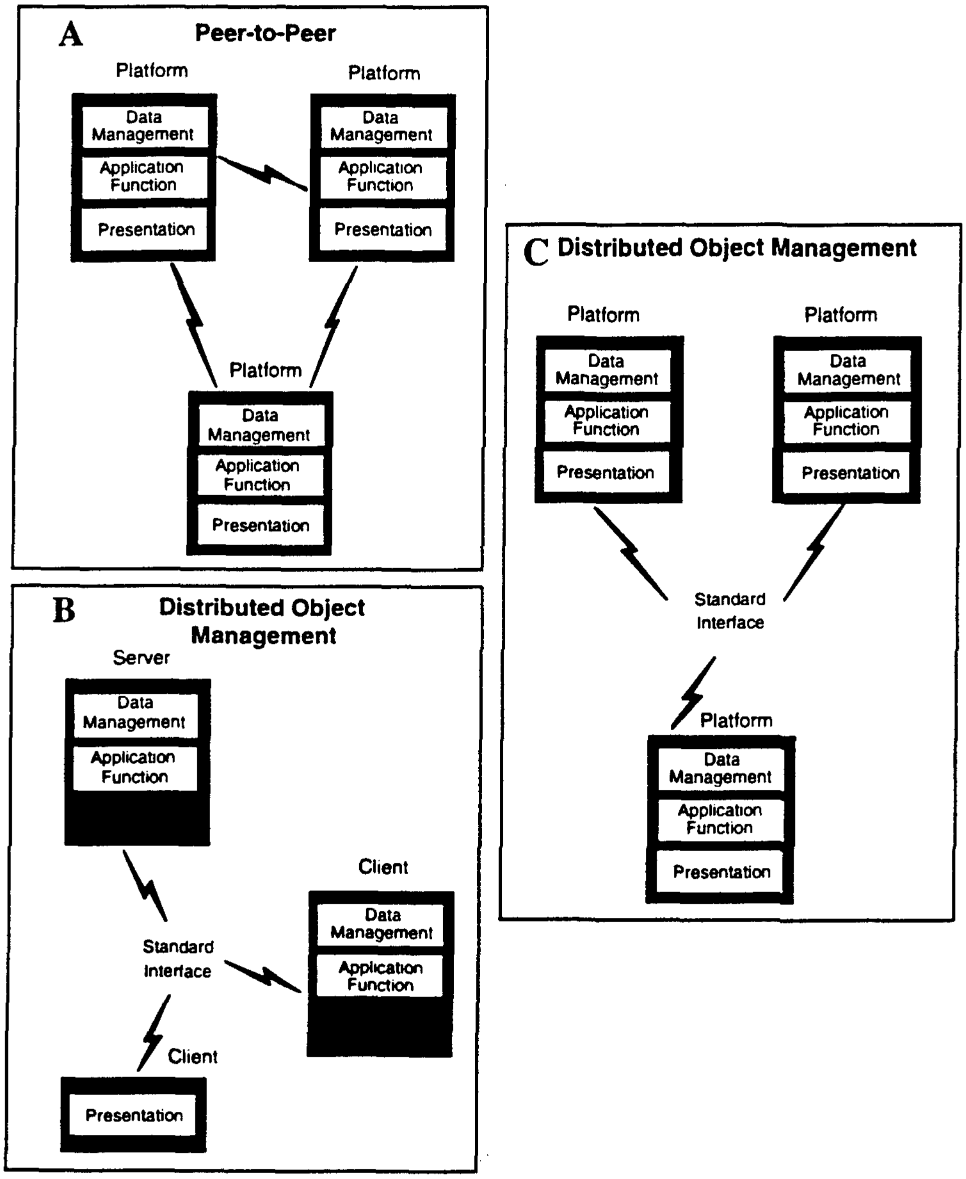


Figure 2-7. Peer-to-Peer and Distributed Object Management Models

The Object Management Group (OMG), a consortium of industry participants working toward object standards, has developed an architecture – the Common Object Request Broker Architecture (CORBA), which specifies the protocol a client application must use to communicate with an Object Request Broker (ORB), which provides services. The ORB specifies how objects can transparently make requests and receive responses. In addition, Microsoft's Object Linking and Embedding (OLE) standard for Windows is an example of an implementation of distributed object management, whereby any OLE-compatible application can work with data from any other OLE-compatible application.

2.3 DATA MANAGEMENT VIEW

The DoD is accomplishing a phased convergence to an open systems environment. This involves the selection of migration systems, defining interim architectures, and performing functional and technical integration. Under the TAFIM, data management services may be provided by a wide range of implementations. Some examples are:

- Mega centers providing functionally oriented corporate databases supporting local and remote data requirements

- Distributed database management systems that support the interactive use of partitioned and partially replicated databases

- File systems provided by operating systems, which may be used by both interactive and batch processing applications.

Data management services include the storage, retrieval, manipulation, backup, restart/recovery, security, and associated functions for text, numeric data, and complex data such as documents, graphics, images, audio, and video. The operating system provides file management services, but they are considered here because many legacy databases exist as one or more files without the services provided by a Database Management System (DBMS).

Major components that provide data management services that are discussed in this section are:

- DBMSs

- Data dictionary/directory systems

- Data security.

These are critical aspects of data management for the following reasons. The DBMS is the most critical component of any data management capability, and a data dictionary/directory system is necessary in conjunction with the DBMS as a tool to aid the administration of the database. Data security is a necessary part of DoD's overall policy for secure information processing.

2.3.1 Database Management Systems

A DBMS provides for the systematic management of data. This data management component provides services and capabilities for defining the data, structuring the data, accessing the data, as well as security and recovery of the data. A DBMS performs the following functions:

- Structures data in a consistent way
- Provides access to the data
- Minimizes duplication
- Allows reorganization, that is, changes in data content, structure, and size
- Supports programming interfaces
- Provides security and control.

A DBMS *must* provide:

- Persistence–The data continues to exist after the application's execution has completed
- Secondary storage management
- Concurrency
- Recovery
- Data definition language/data manipulation language (DDL/DML) – it may be a graphical interface.

2.3.1.1 Database Models

The logical data model that underlies the database characterizes a DBMS. The common logical data models are listed in Figure 2-8. The subsections below discuss each of these database types.

2.3.1.1.1 The Relational Model

A RDBMS structures data into tables that have certain properties:

- Each row in the table is distinct from every other row.
- Each row contains only atomic data; that is, there is no repeating data or such structures as arrays.
- Each column in the relational table defines named data fields or attributes.

Data Model
Relational
Hierarchical
Network
Object-Oriented
Flat File

Figure 2-8. Summary of Data Models

A row of data in a relational database is commonly referred to as a tuple; an example would be a record in a file. An example of a column in a relational table would be a field in a record. A collection of related tables in the relational model makes up a database.

The mathematical theory of relations underlies the relational model – both the organization of data and the languages that manipulate the data. Edgar Codd, then at International Business Machines (IBM), developed the relational model in 1973. It has been popular, in terms of commercial use, since the early 1980s.

2.3.1.1.2 The Hierarchical Model

The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeating information, generally in the child data segments. For example, an organization might store information about an employee, such as name, employee number, department, salary. The organization might also store information about an employee's children, such as name and date of birth. The employee and children data forms a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. If an employee has three children, then there would be three child segments associated with one employee segment. In a hierarchical database the parent-child relationship is one to many. This restricts a child segment to having only one parent segment. Hierarchical DBMSs were popular from the late 1960s, with the introduction of IBM's Information Management System (IMS) DBMS, through the 1970s.

2.3.1.1.3 The Network Model

The popularity of the network data model coincided with the popularity of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. So, the network model permitted the modeling of many-to-many relationships in data. In 1971, the Conference on Data Systems Languages (CODASYL) formally defined the network model. The basic data modeling construct in the network model is the set construct. A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set, hence the multiparent concept is supported. An owner record type can also be a member or owner in another set. The CODASYL network model is based on mathematical set theory.

2.3.1.1.4 The Object-Oriented Model

An object-oriented DBMS (OODBMS) must be both a DBMS and an object-oriented system. As a DBMS it must provide the capabilities identified above in Section 2.3.1. OODBMSs typically can model tabular data, complex data, hierarchical data, and networks of data. The following are mandatory features an object-oriented system should support:

- **Complex objects** – e.g., objects may be composed of other objects.
- **Object identity** – Each object has a unique identifier external to the data.
- **Encapsulation** – An object consists of data and the programs (or methods) that manipulate it.
- **Types or classes** – A class is a collection of similar objects.
- **Inheritance** – Subclasses inherit data attributes and methods from classes.
- **Overriding with late binding** – The method particular to a subclass can override the method of a class at run time.
- **Extensibility** – e.g., a user may define new objects.
- **Computational completeness** – A general purpose language, such as Ada, C, or C++, is computationally complete. The special-purpose language SQL is not. Most OODBMSs incorporate a general-purpose programming language.

2.3.1.1.5 Flat Files

A flat file system is usually closely associated with a storage access method. An example is IBM's indexed sequential access method (ISAM). The models discussed earlier in this section are logical data models–flat files require the user to work with the physical layout of the data on a storage device. For example, the user must know the exact location of a data item in a record. In addition, flat files do not provide all of the services of a DBMS, such as naming of data, elimination of redundancy, and concurrency control. Further, there is no independence of the data and the application program. The application program must know the physical layout of the data.

2.3.1.2 Distributed DBMSs

A distributed DBMS manages a database that is spread over more than one platform. The database can be based on any of the data models discussed above (except the flat file). The database can be replicated, partitioned, or a combination of both. A replicated database is one in which full or partial copies of the database exist on the different platforms.

A major issue with replication is the method of maintaining consistency between the copies of the database. Some database management systems attempt to do this using complex synchronization algorithms (e.g., "two-phase commit" protocols). Many commercial database vendors are offering a simpler form of replication in which a master copy is updated, then changes are propagated to the database copies by a replication server at a later time.

A partitioned database is one in which part of the database is on one platform and parts are on other platforms. The partitioning of a database can be vertical or horizontal. A vertical partitioning puts some fields and the associated data on one platform and some fields and the associated data on another platform. For example, consider a database with the following fields: employee identification (ID), employee name, department, number of dependents, project assigned, salary rate, tax rate. One vertical partitioning might place employee ID, number of dependents, salary rate, and tax rate on one platform and employee name, department, and project assigned on another platform. A horizontal partitioning might keep all the fields on all the platforms but distribute the records. For example, a database with 100,000 records might put the first 50,000 records on one platform and the second 50,000 records on a second platform.

Whether the distributed database is replicated or partitioned, a single DBMS manages the database. There is a single schema (description of the data in a database in terms of a data model, e.g., relational) for a distributed database. The distribution of the database is generally transparent to the user. The term "distributed DBMS" implies homogeneity.

2.3.1.3 Distributed Heterogeneous DBMSs

A distributed, heterogeneous database system is a set of independent databases, each with its own DBMS, presented to users as a single database and system. "Federated" is used synonymously with "distributed heterogeneous." The heterogeneity refers to differences in data models (e.g., network and relational), DBMSs (e.g., Oracle and Ingres), platforms (e.g., VAX and Sun), or other. The simplest kinds of federated database systems are commonly called gateways. In a gateway, one vendor (e.g., Oracle) provides single-direction access through its DBMS to another database managed by a different vendor's DBMS (e.g., IBM's DB2). The two DBMSs need not share the same data model. For example, many RDBMS vendors provide gateways to hierarchical and network DBMSs.

There are federated database systems both on the market and in research that provide more general access to diverse DBMSs. These systems generally provide a schema integration component to integrate the schemas of the diverse databases and present them to the users as a single database, a query management component to distribute queries to the different DBMSs in the federation, and a transaction management component, to distribute and manage the changes to the various databases in the federation.

2.3.2 Data Dictionary/Directory Systems

The second component providing data management services, the data dictionary/directory system (DD/DS), consists of utilities and systems necessary to catalog, document, manage, and use metadata (data about data). An example of metadata is the following definition: a 6-character long alphanumeric string, for which the first character is a letter of the alphabet and each of the remaining 5 characters is an integer between 0 and 9; the name for the string is employee ID. The DD/DS utilities make use of special files that contain the database schema. (A schema, using metadata, defines the content and structure of a database.) This schema is represented by a set of tables resulting from the compilation of DDL statements. The DD/DS is normally provided as part of a DBMS but is sometimes available from alternate sources. In the management of distributed data, distribution information may also be maintained in the network directory system. In this case, the interface between the DD/DS and the network directory system would be through the API of the network services component on the platform.

In current environments, data dictionaries are usually integrated with the DBMS, and directory systems are typically limited to a single platform. Network directories are used to expand the DD/DS realms. The relationship between the DD/DS and the network directory is an intricate combination of physical and logical sources of data.

2.3.3 Data Administration

DoD Directive (DoDD) 8320.1 defines the data administration program for the DoD. Data administration properly addresses the data architecture, which is outside the scope of the TAFIM. We discuss it briefly here because of areas of overlap. It is concerned with all of the data resources of an enterprise, and as such there are overlaps with data management, which addresses data in databases. Two specific areas of overlap are the repository and database administration, which are discussed briefly below.

2.3.3.1 Repository

A repository is a system that manages all of the data of an enterprise, which includes data and process models and other enterprise information. Hence, the data in a repository is much more extensive than that in a DD/DS, which generally defines only the data making up a database.

2.3.3.2 Database Administration

Data administration and database administration are complementary processes. Data administration is responsible for data, data structure, and integration of data and processes. Database administration, on the other hand, includes the physical design, development, implementation, security, and maintenance of the physical databases. Database administration is responsible for managing and enforcing the enterprise's policies related to individual databases.

2.3.4 Data Security

The third component providing data management services is data security procedures and technology measures that are implemented to prevent unauthorized access, modification, use, and dissemination of data stored or processed by a computer system. Data security also includes data integrity (i.e., preserving the accuracy and validity of the data), and protecting the system from physical harm (including preventative measures and recovery procedures).

Authorization control allows only authorized users to have access to the database at the appropriate level. Guidelines and procedures can be established for accountability, levels of control, and type of control. Authorization control for database systems differs from that in traditional file systems because, in a database system, it is not uncommon for different users to have different rights to the same data. This requirement encompasses the ability to specify subsets of data and to distinguish between groups of users. In addition, decentralized control of authorizations is of particular importance for distributed systems.

Data protection is necessary to prevent unauthorized users from understanding the content of the database. Data encryption, as one of the primary methods for protecting data, is useful for both information stored on disk and for information exchanged on a network.

2.4 COMMUNICATIONS VIEW

The Open Systems Interconnection (OSI) model discussed in the following sections is useful as an aid to understanding the elements of successful network communication; however, it should be understood that the OSI protocols contained in the GOSIP standard are no longer mandated for use by Federal agencies. This change resulted from the emergence of Internet Protocol Suite (IPS) standards as the dominant standards for commercial hardware and software, and the relatively smaller number of OSI-compliant products available. Government agencies are now able to select cost-effective, off-the-shelf networking products that implement open standards, such as those developed by the Internet Engineering Task Force (IETF), International Telecommunications Union, and the International Organization for Standardization (ISO). The OSI protocols have been updated and are contained in the Industry/Government Open System Specification, NIST Publication 500-217.

Communications networks are constructed of end devices (e.g., printers), processing nodes, communication nodes (switching elements), and the linking media that connect them. The communications network provides the means by which information is exchanged. Forms of information include data, imagery, voice, and video. Automated information systems (AISs) accept and process information using digital data formats rather than analog formats. Therefore, TAFIM communications concepts and guidance will focus on digital networks and digital services. Integrated multimedia services are included.

The communications view describes the architecture of DoD communications with respect to its geography (Section 2.4.1), discusses the OSI reference model, and describes a general framework intended to permit effective system analysis and planning for DoD (Section 2.4.2).

2.4.1 DoD Communications Infrastructure

The communications infrastructure in the DoD will contain three transport components, local, regional/metropolitan, and global, as shown in Figure 2-9. The names of the transport components are based on their respective geographic extent, but there is also a hierarchical relationship among them.

The transport components correspond to a network management structure in which management and control of network resources are distributed across the different levels.

The local components relate to assets that are located relatively close together geographically. This component contains sustaining base communications assets for the fixed environment and tactical communications assets in the deployed environment. LANs, to which the majority of end devices will be connected, are included in this component. Standard interfaces will facilitate portability, flexibility, and interoperability of LANs and end devices.

Regional and metropolitan area networks (MAN) are geographically dispersed over a large area. A regional or metropolitan network could connect local components at several sustaining bases in the fixed environment or connect theater tactical assets in the deployed environment. In most cases, regional and metropolitan networks are used to connect local networks. However, shared databases, regional processing platforms, and network management centers may connect directly or through a LAN. Standard interfaces will be provided to connect local networks and end devices.

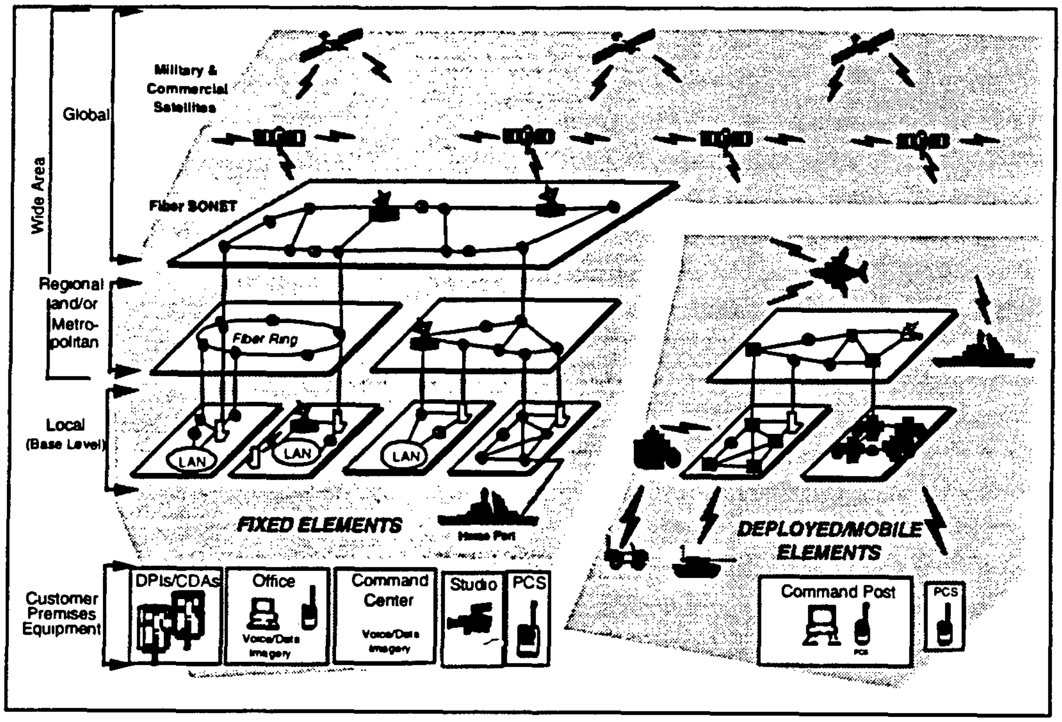


Figure 2-9. Communications Infrastructure

Global or wide area networks (WAN) are located throughout the world, providing connectivity for regional and metropolitan networks in the fixed and deployed environment. In addition, deployed mobile assets, shared databases, and central processing centers can connect directly to the global network as required. Standard interfaces will be provided to connect regional and metropolitan networks and end devices.

The network that will support all DoD data transport requirements is the Defense Information Systems Network (DISN), authorized under the Chairman of the Joint Chiefs of Staff (CJCS) Memorandum of Policy (MOP) 70, dated 5 February 1992. DISN is intended to support National Defense Command, Control, Communications, and Intelligence (C3I) decision support requirements; Corporate Information Management (CIM) functional business areas; and Defense Information Infrastructure (DII) data processing and information transfer services. The DISN will support the DoD's WAN on the global scale. DISN responsibility will extend to the local component in the future.

2.4.2 Communications Models

The geographically divided infrastructure described in Section 2.4.1 forms the foundation for an overall communications framework. These geographic divisions permit the separate application of different management responsibilities, planning efforts, operational functions, and enabling technologies to be applied within each area. Hardware and software components and services fitted to the framework form the complete model.

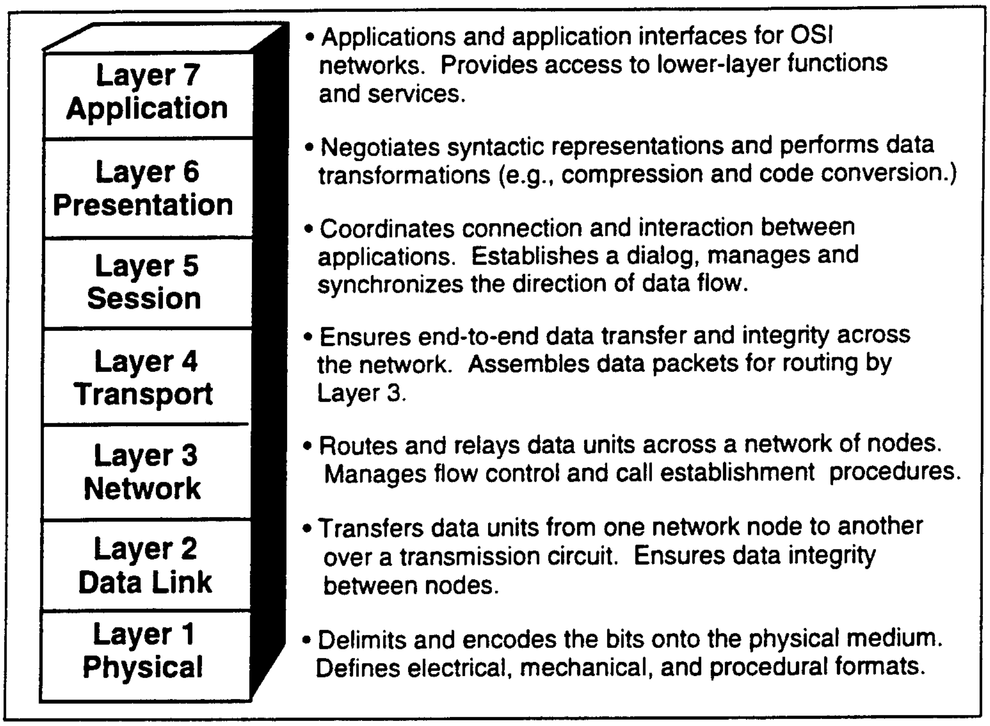
The following sections describe the OSI reference model and a grouping of the OSI layers that facilitates discussion of interoperability issues.

2.4.2.1 The OSI Reference Model

The OSI reference model, portrayed in Figure 2-10, is the model used for data communications in the TAFIM. Although DoD is no longer advocating the full use of OSI protocols, the OSI reference model is a valuable tool for conceptualizing networking requirements and solutions. Each of the seven layers in the model represents one or more services or protocols (a set of rules governing communications between systems), which define the functional operation of the communications between user and network elements.

Each layer provides services for the layer above it. This model aims at establishing open systems operation and implies standards-based implementation. It strives to permit different systems to accomplish complete interoperability and quality of operation throughout the network.

The seven layers of the OSI model are structured to facilitate independent development within each layer and to provide for changes independent of other layers. Stable international standard protocols in conformance with the OSI reference model layer definitions have been published by various standards organizations. Support and mission-area applications, as defined in the DoD Technical Reference Model, are above the OSI Reference Model protocol stack and use its services via the applications layer.



- Applications and application interfaces for OSI networks. Provides access to lower-layer functions and services.
- Negotiates syntactic representations and performs data transformations (e.g., compression and code conversion.)
- Coordinates connection and interaction between applications. Establishes a dialog, manages and synchronizes the direction of data flow.
- Ensures end-to-end data transfer and integrity across the network. Assembles data packets for routing by Layer 3.
- Routes and relays data units across a network of nodes. Manages flow control and call establishment procedures.
- Transfers data units from one network node to another over a transmission circuit. Ensures data integrity between nodes.
- Delimits and encodes the bits onto the physical medium. Defines electrical, mechanical, and procedural formats.

Figure 2-10. Open Systems Interconnection Model

2.4.2.2 Communications Framework

A communications system based on the OSI reference model includes services of all the layers described in the previous section plus the physical transmission media and the support and mission-area applications defined in Volume 2, *Technical Reference Model.* These elements may be grouped into architectural levels that represent major functional capabilities, such as switching and routing, data transfer, and the performance of applications.

These architectural levels are:

- The Transmission Level (below the OSI) provides all of the physical and electronic capabilities, which establish a transmission path between functional system elements (wires, leased circuits, interconnects, etc.).
- The Network Switching Level (OSI layers 1 through 3) establishes connectivity through the network elements to support the routing and control of traffic (switches, controllers, network software, etc.).
- The Data Exchange Level (OSI layers 4 through 7) accomplishes the transfer of information after the network has been established (end-to-end, user-to-user transfer) involving more capable processing elements (hosts, workstations, servers, etc.).
- The Applications Program Level (above the OSI) includes the support and mission-area applications (non-management application programs).

The communications framework is defined to consist of the three geographical components of the DoD communications infrastructure (local, regional, and global) and the four architectural levels (transmission, network switching, data exchange, and application program), and is depicted in Figure 2-11. Communications services are performed at one or more of these architectural levels within the geographical components.

Figure 2-11 shows computing elements (operating at the applications program level) with supporting data exchange elements, linked with each other through various switching elements (operating at the network level), each located within its respective geographical component. Figure 2-11 also identifies the relationship of the Technical Reference Model to the communication architecture.

2.4.2.3 Allocation of Services to Components

The DoD communications infrastructure consists of the local, regional, and global transport components. The services allocated to these components are identical to the services of the application program, data exchange, network switching, or transmission architectural levels that

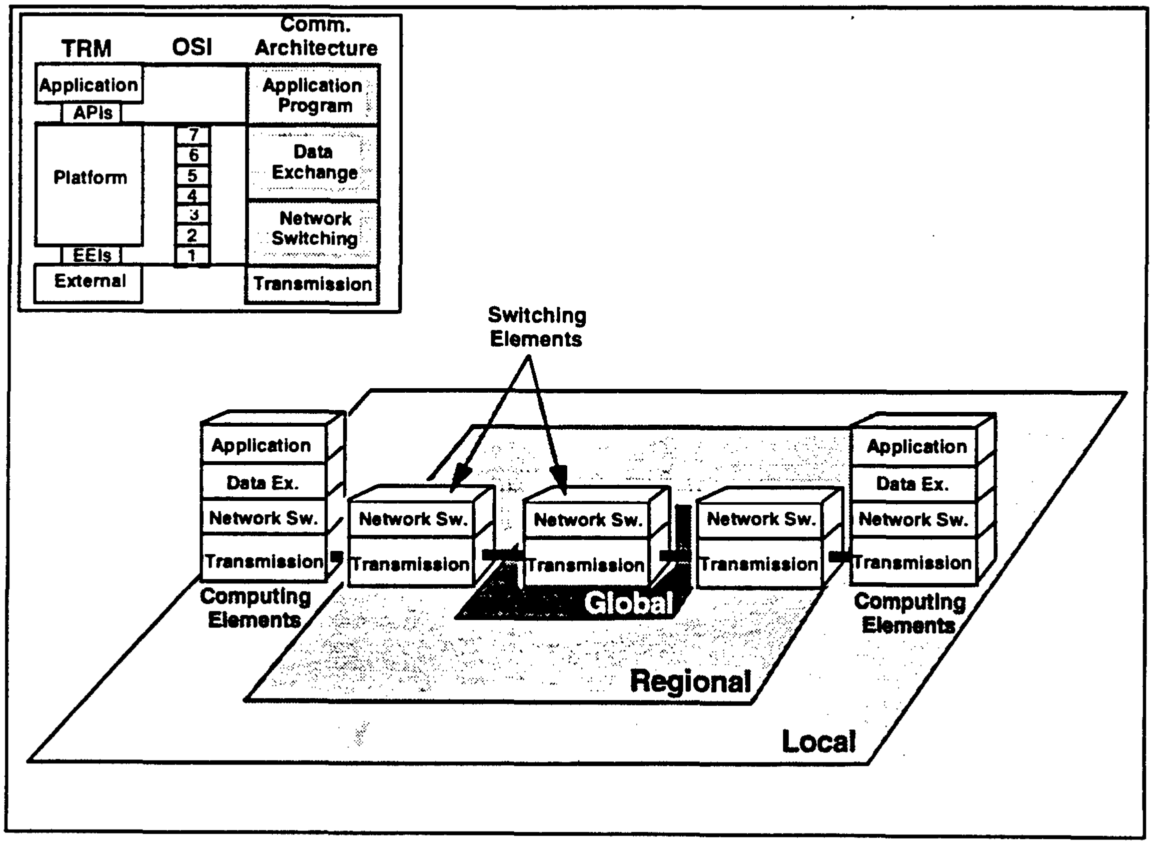


Figure 2-11. Communications Framework

apply to a component. Data exchange and network switching level services are identical to the services of the corresponding OSI reference model layers. Typically, only network switching and transmission services are allocated to the regional and global components, which consist of communications nodes and transmission media. All services may be performed in the local component, which includes end devices, processing nodes, communications nodes, and linking media. Transmission, switching, transport, and applications are all performed in this component.

2.5 SECURITY VIEW

The business of the DoD requires the controlled use of information. Security protection of DoD information systems is discussed in Volume 6 of the TAFIM, *DoD Goal Security Architecture (DGSA)*. The purpose of this section is to provide a brief overview of Volume 6 with a focus on security protection implemented in the information system components. Doctrinal mechanisms, such as physical and personnel security procedures and policy, are discussed in Volume 6 but omitted here.

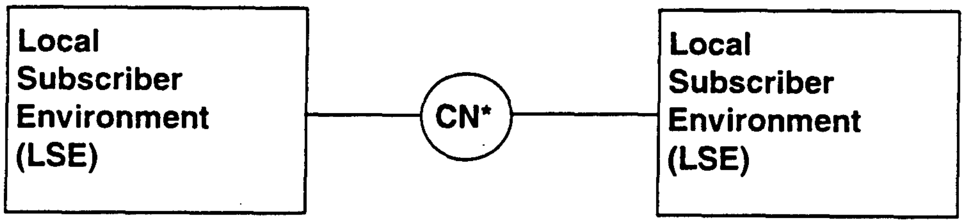
Figure 2-12 depicts an abstract view of an information system architecture, which emphasizes the fact that an information system from the security perspective is either part of a local subscriber environment (LSE) or a communications network (CN). An LSE may be either fixed or mobile. The LSEs by definition are under the control of the using organization. In an open system distributed computing implementation, secure and nonsecure LSEs will interoperate.

2.5.1 Basic Concepts

This section presents basic concepts required for an understanding of information system security within DoD.

2.5.1.1 Information Domains

The concept of an *information domain* provides the basis for discussing security protection requirements. An information domain is defined as a set of users, their information objects, and a security policy. An information domain security policy is the statement of the criteria for membership in the information domain and the required protection of the information objects.


Local Subscriber Environment (LSE)
CN*
Local Subscriber Environment (LSE)

*CN = Communications Network

Figure 2-12. Abstract Security Architecture View

The missions of most DoD organizations require that their members operate in more than one information domain. The diversity of mission activities and the variation in perception of threats to the security of information will result in different information domains within one mission security policy. A specific mission may use several information domains, each with its own distinct information domain security policy.

There must be no security-relevant distinction made among the information objects in an information domain. Members of an information domain may have different security-related attributes. For example, some members might have only read permission for information objects in an information domain, while other members might have both read and write permissions.

Since all information objects in an information domain have the same security-relevant attributes, a user who has read and write permissions in an information domain has those permissions for every information object in the information domain. The term "information object" refers to any type of information.

Information domains are not bounded by information systems or even networks of systems. The security mechanisms implemented in information system components may be evaluated for their ability to meet the information domain security policies.

2.5.1.2 Strict Isolation

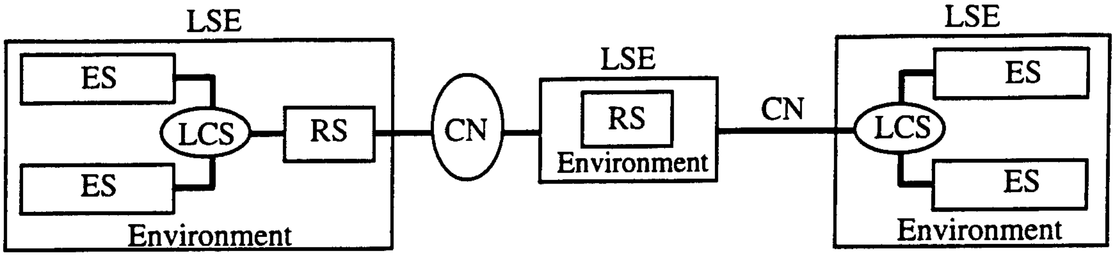
The strategy of "strict isolation" is used to isolate one information domain from another. Information objects can be transferred between two information domains only in accordance with established rules, conditions, and procedures expressed in the security policy of each information domain. Multidomain information objects may be defined for display or printing. A multidomain information object is a defined collection of information objects from multiple information domains.

2.5.1.3 Absolute Protection

The concept of "absolute protection" is used to provide a framework for achieving uniformity of protection in all information systems supporting a particular information domain. It directs attention to the problems created by the interconnection of LSEs that provide disparate strengths of security protection. This possibility is likely because open systems will consist of an unbounded number of unknown heterogeneous LSEs that must be able to interoperate. Analysis related to minimum assurance requirements will ensure that the concept of absolute protection will be achieved for each information domain across LSEs.

2.5.2 Security Generic Architecture View

Figure 2-13 shows the generic architectural view used in Volume 6 to discuss the allocation of security services and the implementation of security mechanisms. This view identifies the architectural components within a LSE. The LSEs are connected by CNs. The LSEs include end systems, relay systems, and local communications systems (LCSs), described below.



Key

CN - communications network

ES - end system

LCS - local communications system

LSE - local subscriber environment

RS - relay system

Figure 2-13. Generic Security Architecture View

- **Relay system** – The component of an LSE, the functionality of which is limited to information transfer and is only indirectly accessible by users (e.g., router, switch, multiplexer, message transfer agent). It may have functionality similar to an end system, but an end user does not use it directly. Note that relay system functions may be provided in an end system.

- **Local communication system** – A network that provides communications capabilities between LSEs or within a LSE with all of the components under control of a LSE.

- **Communication network** – A network that provides inter-LSE communications capabilities, but is not controlled by LSEs (e.g., commercial carriers).

The end system and the relay system are viewed as requiring the same types of security protection. For this reason, a discussion of security protection in an end system generally also applies to a relay system. The security protections in an end system could occur in both the hardware and software.

2.5.3 Security Services Allocation

Security protection of an information system is provided by mechanisms implemented in the hardware and software of the system and by the use of doctrinal mechanisms. The mechanisms implemented in the system hardware and software are concentrated in the end system or relay system. This focus for security protection is based on the open system, distributed computing approach for DoD information systems. This implies use of commercial common carriers and DoD-owned common-user communications systems as the CN provider between LSEs. Thus, for operation of end systems in a distributed environment, a greater degree of security protection can be assured from implementation of mechanisms in the end system or relay system.

However, CNs should satisfy the availability service to promote satisfaction of appropriate security protection for the information system. This means that CNs must provide an agreed level of responsiveness, continuity of service, and resistance to accidental and intentional threats to the communications service availability.

End systems may not need to interoperate with others, but may need to accommodate multiple security domains processing simultaneously.

Implementing the necessary security protection in the end system occurs in three system service areas. They are operating system services, network services, and system management services.

Most of the implementation of security protection is expected to occur in software. The hardware is expected to protect the integrity of the end system software. Hardware security mechanisms include protection against tampering, undesired emanations, and cryptography.

2.5.3.1 Operating System Services

A "security context" is defined as a controlled process space subject to an information domain security policy. The security context is therefore analogous to a common operating system notion of user process space. Isolation of security contexts is required. Security contexts are required for all applications (e.g., end user and security management applications). The focus is on strict isolation of information domains, management of end system resources, and controlled sharing and transfer of information among information domains. Security-critical functions are isolated into relatively small modules that are related in well-defined ways.

The operating system "separation kernel" will maintain the required isolation. The separation kernel will use the protection features of the end system hardware (e.g., processor state register, memory mapping registers) to maintain strict separation among security contexts by creating separate address spaces for each of them. Untrusted software will use end system resources only by invoking security-critical functions through the separation kernel. Security-critical functions perform inter-security context (i.e., inter-information domain) operations. Most of the security-critical functions are the low-level functions of traditional operating systems.

2.5.3.2 Network Services

Two basic classes of communications are envisioned for which distributed security contexts may need to be established. These are interactive and staged (store and forward) communications.

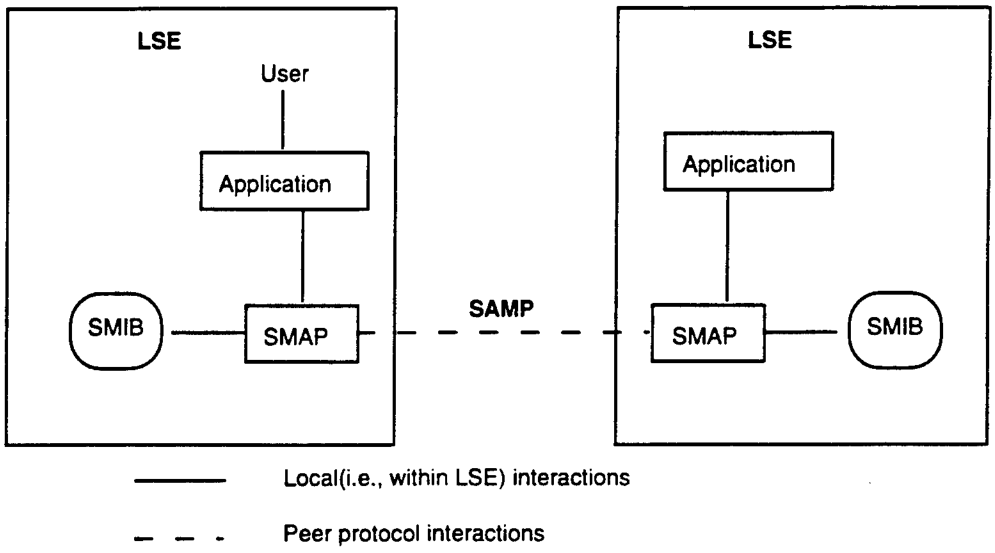
The concept of a "security association" forms an interactive distributed security context. A security association is defined as the totality of communication and security mechanisms and functions to extend the protections required by an information domain security policy within an end system to information in transfer between multiple end systems. The security association is an extension or expansion of an OSI application layer association. An application layer association is composed of appropriate application layer functions and protocols plus all of the underlying communications functions and protocols at other layers of the OSI model. Multiple security protocols may be included in a single security association to provide for a combination of security services. However, a security association can only be established within the same information domain; inter-information- domain security associations are not allowed.

For staged delivery communications (e.g., e-mail), use will be made of an encapsulation technique (termed "wrapping process") to convey the necessary security attributes with the data being transferred as part of the network services. The wrapped security attributes are intended to permit the receiving end system to establish the necessary security context for processing the transferred data. If the wrapping process cannot provide all the necessary security protection, interactive security contexts between end systems will have to be used to ensure the secure staged transfer of information.

2.5.3.3 System Security Management Services

Security management is a particular instance of general information system management functions as discussed in Volume 2. Information system security management services are concerned with the installation, maintenance, and enforcement of information domain and information system security policy rules in the information system intended to provide these security services. In particular, the security management function controls information needed by operating system services within the end system security architecture. In addition to these core services, security management requires event handling, auditing, and recovery. Standardization of security management functions, data structures, and protocols will enable interoperation of security management application processes (SMAPs) across many platforms in support of distributed security management. Areas for security management standardization are described in Volume 6, *DoD Goal Security Architecture (DGSA)*.

SMAPs, using information in the information base, will be used to establish the required security contexts for interactive communications among distributed platforms operating in various information domains simultaneously. System SMAPs will also be used to provide the security protection of store-and-forward communications in which the requisite security contexts cannot be handled within the message. The end system will establish a security association by using a SMAP, a security association management protocol (SAMP), and information in the security management information base (SMIB). Figure 2-14 shows the general relationship of the processes and protocols involved in establishing a security association for interactive communications among distributed end systems.


LSE
User
Application
SMIB
SMAP
SAMP
LSE
Application
SMAP
SMIB
Local(i.e., within LSE) interactions
Peer protocol interactions

Key
LSE - Local subscriber environment
SAMP - Security association management protocol
SMAP - Security management application process
SMIB - Security management information base

Figure 2-14. Architectural Components Involved in Establishing a Security Association for Interactive Communications

3.0 DESIGN GUIDANCE

The architectural concepts discussed in Section 2 are needed to realize the long-term DoD IM vision of an open distributed computing environment. In the near term, new information systems will be engineered and integrated into an environment that includes legacy and migration systems. Many legacy and migration systems use non-standard and proprietary approaches. This complicates the integration of new systems designed for an open systems environment. This chapter focuses on near-term application of the architecture concepts. It includes guidance for integrating open systems, legacy systems, and migration environments. The guidance presented in this section will evolve over time based on the availability and maturity of technology. The guidance in this section supplements the concepts in Volume 2.

3.1 GUIDANCE FOR DESIGNING ARCHITECTURES

An architecture is a set of components and a specification of how these components are connected to meet the overall requirements of an information system. The components of an architecture provide implementations of the reference model services relevant to a specific system. The following are guidelines for designing specific architectures given the Technical Reference Model and the model of information system architecture:

- An architecture will contain components to implement only those reference model services that it requires.

- Components may implement one, more than one, or only part of a service identified in the reference model.

- The components should conform to the profile standards that are relevant to the services they do implement.

The following is a general procedure for designing a specific architecture given the guidelines above:

- Perform requirements analysis

- Make service allocations

- Select components

- Evaluate.

3.2 COMPUTING MODELS

This section presents guidance on the computing models discussed in Section 2.2.

3.2.1 Client/Server Model Design Analysis and Guidance

The objective computing environment is a distributed computing environment based on open systems principles and public standards (e.g., Federal Information Processing Standards [FIPS] and ISO). In this environment, services will be provided by servers distributed to processing nodes throughout a network. Services will be distributed based on attention to survivability, efficiency, and functional requirements. As DoD migrates to a distributed computing environment, many different types of client/server capabilities will be established. These will support, among other things, the initial implementation of subject area databases, access to data managed by migration systems from open system platforms, and interactive applications that are distributed to a POSIX platform and accessed via a standards-compliant graphical user interface (GUI). The following are some guidelines related to the client/server model:

- Provide client processes with a high-level interface with as few details about the underlying communication and processing details as possible. An example interface is that provided by the CORBA. Design the client/server mechanism so that the location of the server can change without impacting the client application.

- Isolate the client application from the details of the interprocess application. This would allow details of the communication mechanism to change without affecting the client application. This would also allow requests for services to be provided by both local and remote servers, and the client would be insulated from the details.

- Design subject area databases to provide users and client processes with the capability to query data without knowing where or how the data is physically stored. Use distributed processes to provide record level access to mainframe migration databases. Until products are readily available that comply with an open standard, the preferred method is through a de facto standards-based implementation of the client/server model. This will require implementation of a de facto standard on the appropriate mainframe platform. End-user workstations that require this type of access will also require a de facto standard implementation. This will give distributed users and client processes access to both query and update data managed by migration mainframe databases.

- Use other implementations of the client/server model achievable today, including access to file and print servers provided through a network file service. This can provide users with remote access to files and network printers. These services should be provided in such a way that there is a migration path to open standards when they become available.

To summarize, the client/server model provides a flexible framework for designing and maintaining distributed processing applications. New application-enabling technology, such as computer-aided software engineering (CASE) and GUIs, focus on this model. Some general advantages of the model are:

- Readily supports the open systems concepts of portability, interoperability, scalability, modularity, and flexibility
- Allows for the continued use of existing capital investments such as PCs, LANs, minicomputers, and mainframes
- Enables data sharing among many different applications
- Accommodates special function hardware or software that does not have to be duplicated
- Allows data and processing to be distributed to the appropriate organizational level
- Supports centralized control and security of data
- Supports survivability through the management and distribution of redundant data and processes
- Supports consistent user interfaces across applications.

3.2.1.1 Guidance on the Multitiered Architecture

Industry evolved from single-tiered architectures on mainframes to two-tiered architectures (often still on mainframes) because there was a recognized need to separate data from applications. In the single-tier architecture, each application had its own data in its own files and there was little, if any, opportunity for application A to share application B's data. Once the data was separated from the applications, which has often happened in actual implementations of the two-tiered client/server architecture, the data became a resource that could be used across multiple applications. Inconsistent data sets were eliminated (in concept at least), concurrent access to data was allowed, integrity constraints on data were supported, and data was protected.

The multitiered approach is an extension of the two-tiered approach. By now separating the user interface (i.e., the presentation layer) from the rest of the application (remember, data has already been separated out), the functional code of the application can be turned into the elements of a reusable library of functional routines. Furthermore, those routines can begin to be executed on networked platforms (possibly remote from the user platform); the elements of the presentation layer can also be turned into a library of reusable elements, and the user interface can be replaced with a new interface without having to rewrite the entire application.

The multitiered approach will allow migration of legacy systems to modular systems and taking advantage of the benefits mentioned above. However, the separation of the application logic and presentation layers may exclude certain COTS systems that would otherwise offer significant benefits – in particular, there are COTS products (two-tier) that offer the benefit of porting their clients to multiple platforms with a simple recompilation. There are also products (again two-tier) that offer very powerful data access and manipulation capabilities across multiple data servers (like cross-database joins) that would otherwise have to be coded and maintained as part of the "functional" code.

New system and migration system architectures should carefully consider the relative advantages of specific COTS products as well as the two-tiered and multitiered approaches based on the specific system's requirements. No single solution will meet the needs of all DoD systems.

3.2.2 Guidance for Other Computing Models

The compelling advantages of the client/server model must be weighed against many potential disadvantages. The requirements of some environments cannot withstand the potential disadvantages of a migration to client/server in the near term. When considering migration from a centralized or mainframe environment, the following disadvantages of the client/server model should be considered:

- Client/server computing is heavily dependent upon the reliability and performance characteristics of the network.
- Security and data integrity requirements are more complicated than when processing is performed on a single (e.g., mainframe) platform. Access to servers and services must be limited to authorized clients. Each client also must be able to select a specific server and be assured that only this server gets access to its data during callbacks. The sender of messages must be assured that messages are neither read nor distorted by other parties. A secure message requires an authentication service and an encryption technique. The message protocol must be able to support the needed security services as determined at runtime.
- Client/server implementations usually entail the integration of a more diverse set of products (than for mainframe-based implementations), increasing the integration effort, complexity, and risk.
- System management, administration, performance monitoring, fault isolation, and correction are more difficult. This is due principally to a lack of tools, a more complex environment, and the interaction of diverse components.
- Development and maintenance staffs that have been working in mainframe environments often do not possess the necessary skills to implement and maintain client/server systems.
- The expected cost saving through the use of low-cost commodity processors may be offset by increased administration costs and the need for highly qualified support personnel.
- Initial client/server implementations often take longer and cost more than expected due to lack of familiarity with the development process and tools.

- It is difficult to replicate problems at a vendor or other central support site. Client/server systems are often a combination of COTS products, protocols, local configuration parameters, and developed software that result in a system with many unique attributes.

- The security implications are not as well understood in comparison with those for mainframe solutions.

When several of these disadvantages are concerns, a client/server implementation may not be the best solution. A mainframe-based approach such as the master-slave or three-tiered model may be more appropriate.

With respect to other models, the following considerations should be addressed. The peer-to-peer approach is often considered a superior alternative to the client/server approach, providing client/server capabilities with the advantage that an application can be processed on any computer in the network – wherever the computing resources are available. However, there are technical challenges associated with the peer-to-peer model that have not been overcome to date, and very few implementations of this model are commercially available. The major value of this type of design approach is that it enhances system availability. This approach could be used to provide system availability for implementations of other computing models. It could provide a hot standby for a centralized system or hot standbys for servers in a client/server implementation. Some types of servers that would benefit from redundancy are naming, communications, and database.

The distributed object management model can be considered as a special case for the client/server or peer-to-peer model, and some consider it superior to the client/server model because of its attempt to provide a cleaner, simpler interface between systems. This model significantly reduces the number of interfaces required among interoperating platforms. It requires only one interface for all platforms as opposed to one interface for each pair of platforms. This is a significant advantage. In addition, once a system has sent a message to another system, the sending system is not required to cease all processing while awaiting a response.

3.3 DATA MANAGEMENT

The near-term goals for improving data management in the DoD are focused on consolidation and interoperation. Consolidation is being carried out primarily in the context of existing applications, resulting in the need to merge databases of similar applications into a single migration system. The surviving system will not necessarily be upgraded to meet target architectural standards. As a result, many existing data management technologies will still be in use at the end of the near-term period. Longer term, the strategic combination of data management components and the standards in Volume 2 of the TAFIM will allow DoD to evolve to an open systems environment. This can be achieved through the identification of flexible data management components and the implementation of systems intended to enhance DoD-wide interoperability.

3.3.1 Guidance on DBMSs

The alternative to a DBMS is a flat file system. The primary advantages of a DBMS are data independence and controlled redundancy. In addition, DBMSs provide capabilities for defining a database through a schema, querying and manipulating the data, concurrency control, and systematic backup and recovery. Flat file systems provide none of these capabilities. A DBMS is preferable to a flat file system.

Flat files might be chosen over DBMSs in the rare event that the application and the file require a very high level of performance (a flat file system does not carry the overhead burden of a DBMS) and very little maintenance.

3.3.1.1 Guidance on Database Models

The RDBMS has become the DBMS of choice over the hierarchical and network model DBMSs. There are several reasons for this. Because of the complexity of the structures involved in the hierarchical and network models, the manner of accessing the data, and the implementation of the structures, both the hierarchical and network models are considered significantly more difficult to manage than the relational model. With the hierarchical and network DBMSs, the application programmer must specify the navigation path for reaching data, e.g., the complete hierarchical path for reaching a data item, as opposed to just the attribute name in the case of an RDBMS. In addition, physical pointers are used in hierarchical and network DBMSs to represent the parent-child and owner-member relationships, respectively. These physical pointers present a difficult challenge in pointer maintenance. The tabular representation of data in the relational model and the relational algebra and calculus languages used to interact with relational databases are considered much simpler than network and hierarchical systems, and application programming and maintenance are much easier. In addition, the performance problems of the early RDBMSs have been overcome, and it has been shown that data that can be modeled as hierarchies and networks can also be modeled as relations. Therefore the RDBMS is recommended over the network and hierarchical DBMSs.

Many experts say we are at the beginning of a new generation of DBMSs – the object-oriented and extended relational. An extended relational DBMS is a relational DBMS with some object-oriented features, generally class inheritance, complex data, or large objects (such as text and graphics). RDBMSs are excellent for conventional or business data, such as personnel and payroll, but not for non-conventional data and applications. Examples of applications requiring non-conventional data are CASE, computer-aided design (CAD), computer-aided manufacturing (CAM), and expert systems. These new applications require multiple data types and the ability to represent complex relationships among the data. In addition, the new applications require modeling power, long and design transactions, and version and configuration management. OODBMSs are being designed and manufactured to meet these needs. A criticism of OODBMSs is that they are reminiscent of network DBMSs with their pointers to implement relationships. However, OODBMSs use logical, not physical pointers, and the difficulty of pointer maintenance present in network DBMSs does not apply to OODBMSs. Another criticism of OODBMSs is that there is no formal theory behind the model, as with the relational model.

However, there are formalisms involved, such as generalization (class/subclass/inheritance), aggregation (objects composed of other objects), and object identity. Standards are not final in this area, and as such the OODBMS presents some risk (e.g., in portability).

OODBMSs fill a gap in the data management area with respect to non-conventional applications that are becoming more prevalent. Most of the commercial OODBMS products are oriented to client/server environments. Procurement of an OODBMS is recommended when the need to support non-conventional applications is present.

3.3.1.2 Guidance on Distributed DBMSs

Replicated databases are recommended when survivability, availability, low transmission cost, and quick response time are important. Replicated databases enhance survivability because if one copy is destroyed in a disaster, other copies are available at other locations. Availability is enhanced for similar reasons. Replicated copies can be located close to the users, so transmission costs are less. For the same reason, response time should be less.

A disadvantage of fully synchronized replicated databases is that updates are expensive because of the need to maintain complete consistency between copies of the database. This form of replication can also result in the inability to update the database if one of the copies is unavailable due to network or system problems. Therefore this form of replication is not recommended when there are frequent database updates. The use of delayed replication servers is recommended except in the rare cases where absolute consistency is required.

A disadvantage of replicated databases is that updates are expensive because of the need to synchronize the updates of the copies of the database. Therefore, this distributed DBMS approach is not recommended when there are frequent database updates.

Partitioned databases are recommended when: 1) there is a high locality of reference – data at a site is used most by local users and infrequently by remote users; 2) retrieval costs are a concern – these costs are lower; and 3) update costs are a concern – these costs are also lower.

3.3.1.3 Guidance on Distributed Heterogeneous DBMSs

Many DBMSs provide gateways to databases managed by other DBMSs. Gateways are recommended when an organization has standardized on one DBMS, but there is other data, either legacy or in another organization, that the organization needs to access. Gateways are generally limited – that is from one DBMS to one or two other DBMSs without a general solution to federating databases. Gateways are usually an option when procuring a DBMS (e.g., a gateway to DB2 when Sybase RDBMS is purchased). Gateways are recommended when there is a specific need to access a second DBMS using the organization's standard DBMS.

Federated database systems are usually sold by third parties for use in integrating databases managed by different vendor DBMSs. For example, a federated database product might be sold that integrates relational, hierarchical, network, and object-oriented DBMSs. These products attempt to present a general solution for integrating data. Typically, a common data model (e.g., entity-relationship, object-oriented, or relational) is used to develop schemas of the shared data from all the databases. A user sees schemas or views in this common data model and accesses any participating database using the common data language that is provided. In this approach a user is not burdened with having to know the data models and languages of all the participating DBMSs. Federated database systems are an acknowledgment that different autonomous groups often make different decisions on which data model (e.g., relational versus object-oriented) or DBMS to use. Yet the results of the different choices may well have to interoperate. This applies to integrating legacy and migration databases, as well as the integration of different open system databases. Federated database systems are recommended for the interoperability of autonomous database systems – legacy, migration, and open.

3.3.2 Guidance on Data Dictionary/Directory Systems

All DBMSs procured should include an integrated DD/DS.

3.3.3 Guidance on Data Administration

The 8320 series of DoDD provides detailed guidance on data administration.

3.3.4 Guidance on Data Security

Commercially available data management components typically provide integrity and availability services. The integrity services are used to maintain internal database consistency, while availability services control concurrent access to database resources. Some degree of identity-based confidentiality protection is also provided by being able to specify the data management commands that certain users are allowed to execute with respect to certain database objects. To protect the confidentiality of classified or unclassified-but-sensitive data, use of a trusted database management system may be recommended.

DBMSs that have undergone the National Computer Security Center (NCSC) evaluation process are recommended for secure environments. It is important to note that the NCSC evaluates DBMSs based on their ability to enforce a defined confidentiality policy. Enforcement of an integrity and availability policy is not part of the current evaluation requirements, although the National Institute of Standards and Technology (NIST) test suite for compliance with the American National Standards Institute (ANSI) SQL standard does test for the support for those features that are part of the current ANSI SQL standard. Since the evaluation of trusted DBMSs is a new procedure, additional guidance in this area may be forthcoming.

In the near term, several options exist: Treat the database management system as an untrusted component operating on a trusted operating system; begin utilization of relational database management systems that are in the evaluation process, while supplementing systems with other controls to compensate for the low assurance rating; or begin using a combination of trusted database management system and trusted operating system capabilities to achieve higher assurance for label-based confidentiality policy. The first approach allows the application to use the full suite of commercial software but does not provide support for needed security functionality. The other two approaches provide confidentiality support to various levels of assurance but may require a sacrifice of application functionality because the full suite of commercial software may not be compatible with the selected trusted products.

Use of client/server architectures in a distributed configuration for data management functions is highly appropriate but may not be fully supported by the current suite of trusted database management system products. In addition, these systems provide limited or no support for distributed data management functions with enforcement of confidentiality. If the data management capability does support operation in a distributed configuration, the other non-data-management components must also be considered. This is required because the point of exposure to security vulnerability involves each system component and the communication system, since requests and responses must be protected as they travel through the distributed system.

3.3.5 Transition Components

In the environment of the future, there will be a DoD information architecture with standardized components and local nonstandard systems. The data architecture will include the DoD data model, standard database structures, standard data elements, and procedures. The current environment of legacy systems will need to migrate to the new environment. In transitioning to this new environment, many of the data management components discussed in Section 2.2 will be used. In the near future, RDBMSs will begin to proliferate. Database gateways and federated systems will exist to link the RDBMSs with legacy data in flat files and hierarchical and network databases. Some OODBMSs will begin to appear, and federated systems will also link them with the RDBMSs and legacy systems. Distributed DBMSs will be used to provide survivability, availability, and the placement of data closer to the users.

3.4 COMMUNICATIONS

The DISN will evolve to become the common, worldwide communications infrastructure. It will provide integrated data, voice, video, and imagery services with connectivity for command and control systems, intelligence systems, and business systems. In the future, DISN will provide consolidated communications services that efficiently satisfy DoD connectivity requirements.

The DISN will provide or facilitate the following capabilities:

- User logon to any number of computers from any number of locations
- Communications at all levels of classification
- Support to real world events as users change locations and network nodes and end devices change locations
- Dynamic network management to ensure that essential communications receive priority, that the network adjusts to the addition or deletion of communications nodes and links, and that messages are routed to intended recipients regardless of their actual location

- Connectivity or gateways to other federal communications networks and commercial networks

- Mobile communications on land, in the air, or at sea that will incorporate wireless service to extend connectivity to mobile users

- Military-unique requirements, including precedence and preemption

- Linkages to civil and commercial elements

- Theater communications that may be rapidly deployed, robust, and reliable, and that support all the military-unique requirements.

3.4.1 Communications Design Guidance

The following guidance is provided on communications:

- All communications requirements will be fulfilled using communications services and networks that adhere to the DISN architecture and the guidance provided in TAFIM Volume 2.

- Communications systems will provide compatibility with the Defense Message System (DMS).

- Where it is consistent with functional requirements, information systems will rely on DISN rather than providing their own communications capability.

3.4.2 Transition Elements

The current global communications network is a loosely connected collection of legacy and migration systems. In the near term, transition elements, such as application and network gateways, will provide greater connectivity and increased availability. In the mid to long term, transition elements will be phased out as all networks adopt accepted open systems elements. For the foreseeable future, the global communications network will remain a network of networks. The communications network will appear to be global. However, it will actually consist of a large number of smaller networks interconnected by gateways.

3.4.2.1 Multiple Protocol Networks

In a network of networks configuration, terminals, personal computers, and workstations will be directly connected to a local area network. The local area networks will be connected via a gateway to a metropolitan, regional, or wide area network. The metropolitan, regional, or wide area networks provide connectivity with other local area networks. In the near term, it is likely that many of these devices will use different communication protocols. Connectivity can be accomplished with multiple protocol networks. The multiple protocol network should not, however, be viewed as a long-term architectural solution. It provides connectivity directly among local networks with compatible protocol profiles but does not necessarily provide interoperability between local networks.

3.4.2.2 Techniques for Achieving Interoperability

Many existing systems are designed with proprietary protocols (e.g., IBM's prevalent System Network Architecture [SNA]). Bringing such systems into compliance with accepted open systems standards to achieve systemwide interoperability may be accomplished in several ways. Methods of achieving interoperability between different protocol systems include:

- **Total Adoption** – Changing all elements throughout the system at all architectural levels defined in Section 2 to operate with the specified standard protocol set (i.e., making the standard "native").
- **Conversion/Application Gateway** – Employing a gateway as an interface between two disparate network protocols to convert one to the other (e.g., IBM's SNA to TCP/IP). The gateway process involves performing all the functions of each protocol set (from the Network Switching through the Application Program levels) to retrieve the original application layer data and commands and then reintroducing it to the second protocol set. The application gateway is described further in Section 3.4.2.3.
- **Adaptation** – Substituting a different set of top layer protocols to ride on a lower layer standard (e.g., selecting a application set other than Telnet, File Transfer Protocol (FTP), or Simple Mail Transfer Protocol (SMTP) to work with TCP/IP). The process would incorporate several upper level protocol sets at a single processing location. It permits transmission with existing lower level protocols but with different Application layer protocols.
- **Multiple Stacks** – Equipping processors (workstations, servers, or mainframes) with several complete protocol sets. This gives the user the ability to enter all networks for which the processor retains a compatible protocol.
- **Encapsulation** – Wrapping the carrying protocol around the original protocol (e.g., TCP/IP wrapped around SNA) may be performed by a router. Encapsulation allows the carrying protocol to appear transparent (the original protocol enters and exits a network unaltered), permitting the original network elements to continue to operate without alteration.

3.4.2.3 Application Gateways

Application gateways may be used as a transition solution for the interconnection of two networks that adhere to different sets of standards. For example, application gateways could be used to connect a legacy network that supports military standard protocols to a network that is compliant with accepted open systems protocols. The gateway machine would be connected to both communications networks and support different connections on each end to enable the transfer of files. Applications that need to transfer files between the networks would use this application gateway and the gateway machine.

Application gateways are not beyond the current state of technology. There are platforms that have software installed to allow access by multiple protocols. For example, there are platforms that allow both FTAM and FTP access and others that support multiple protocols of electronic mail or messages.

Application gateways are not a solution and are not available for all application areas. If the protocols of the two applications are not sufficiently similar, the application gateway will not be viable. It will not be able to offer the functionality and robustness required. If either of the two applications does not have a significant market share, vendors will be hesitant to build an application gateway. That is, the gateway will not be commercially available.

3.4.2.4 Network Gateways

The global network will be made up of networks or communications links based on many different technologies. There will be links that are based on radio waves, optical beams, and fiber, coaxial, and copper cables. Communications is accomplished or optimized by using protocols that are uniquely matched to the network's or link's technology. The network gateway operates at the Network Switching level and is used to create the connection between links or networks based on different Network Switching level protocols. Thus, the network gateway is used to permit the compatible use of various technologies in the Transmission and Network Switching levels rather than facilitate Application level interoperability.

The network gateway performs packet translation. A number of protocols, which adhere to de jure and de facto standards, are based on a data packet. The network gateway understands how each of the protocols positions data within the packet. As it moves the packet from the source network to the destination network, it translates the data within the packet. Hence, each network only receives packets of the expected format. Network gateways that interconnect networks of different technologies and perform translation are viable and commercially available.

3.5 SECURITY PROTECTION GUIDANCE

3.5.1 Introduction

Volume 6 provides guidance to information system designers and implementors in the form of security principles and target security capabilities. The intent of this section is to provide a brief overview of selected guidance highlights of Volume 6 for individuals who are not information system security specialists. This section also contains two tables with general information about location of security services in the various OSI layer protocols and about appropriate mechanisms used to provide required security services. The source of these two tables is ISO 7498-2.

The guidance in Volume 6 is general because various mechanisms or combinations of mechanisms or services may be used or necessary to satisfy the requirements of the security policy for the information domain(s) handled in any particular information system. A wide variety of specific implementations, dictated by mission and threats, will be needed. The information system designer and implementor will need to work with the designated approving authority of the information system to identify the required level of protection and suitable mechanisms and services. In addition to mechanisms that may be implemented in the hardware and software of the information system, mechanisms that are doctrinal (i.e., physical, administrative, and personnel) will also be used to achieve the necessary level of security protection for the information domains handled by the information system.

The following factors should be considered in determining appropriate security mechanisms:

- Strength of security mechanisms
- Characteristics of security mechanisms
- Cost of security mechanisms
- Performance penalties.

As described in Section 2.5, implementation of security services and mechanisms may be allocated to the various components within a LSE and to the CN. The guidance focuses on implementation of security service mechanisms in end systems (or relay systems). Since end systems and relay systems are viewed as requiring the same kinds of security protection, guidance pertaining to an end system generally also applies to a relay system. The security protection provided in an end system will be implemented in both the hardware and software.

A minimum assurance analysis should be performed to satisfy the requirements of absolute protection as defined in Volume 6.

3.5.2 Guidance for End Systems and Relay Systems

A variety of choices exist for implementations of security mechanisms between the hardware and software portions of an end system or relay system.

3.5.2.1 Hardware Guidance

Implementations in hardware should:

- Enforce isolation of software functions by use of protected paths between users and applications and between application functions
- Ensure software and hardware integrity by use of anti-tampering and unwanted radiation devices/techniques
- Ensure availability by use of fault tolerant and fault detecting architectures.

Cryptographic mechanisms should be used for maintaining strict isolation for information in transfer between end systems. The cryptographic devices should be sufficiently flexible to support requirements of different information domains. It may be necessary to use multiple cryptographic devices.

3.5.2.2 Software Guidance

A software architecture built around trusted and untrusted software components is recommended. Trusted software should be used for security-critical functions, including exchanges between information domains. Trusted software must be evaluated and maintained under strict configuration management. Untrusted software should only be able to invoke functions through the use of trusted software. Nonetheless, it is recommended that untrusted software be obtained from reliable sources, tested before use, and be subjected to integrity safeguards to preclude its modification. Configuration management should also be applied to it. Security protection is provided for guidance in three service areas of the end system software:

- Operating system services
- Network services
- System management services.

3.5.2.2.1 Operating System Services

The recommended end system security architecture relies upon an engineering approach that seeks to isolate security-critical functions into relatively small modules that are related in well-defined ways. Security-critical functions should generally provide commonly used, low-level operating system functions. This is considered consistent with commercial operation system vendors' design and implementation strategies. Volume 6 identifies some security-critical functions. Prototyping and experimentation is also needed to identify other software functions that need to be handled as security-critical.

3.5.2.2.2 Network Services

Communications protocols are to be used for implementing security protection mechanisms for inter-end-system information transfers within the same information domain. The allocation of security services to the various OSI layers is shown in Figure 3-1. As shown, no security services are allocated to OSI layer 5, and no specific services are allocated to layer 6. It also needs to be noted that all services are allocated for possible implementation in OSI layer 7, the application layer. However, the implementation may not be in OSI layer 7, but rather in the application process using the communications services. Details of the rationale for allocation of the security services to the OSI layers are contained in ISO 7498-2. Security protocols relevant to the layers shown in Figure 3-1 are given in Volumes 6 and 7.

Some lower layer security protocols can multiplex several security associations between the same end systems. It is not expected, however, that multiplexing for information systems handling different information domains simultaneously will be acceptable to a designated approving authority.

Service	Layer 1	Layer 2	Layer 3	Layer 4	Layer 7**
Authentication: Peer Entity and Data Origin	N	N	Y	Y	Y
Access Control	N	N	Y	Y	Y
Confidentiality: Connection Oriented	Y	Y	Y	Y	Y
Confidentiality: Connectionless	N	Y	Y	Y	Y
Confidentiality: Selective Field	N	N	N	N	Y
Traffic Flow Confidentiality	Y	N	Y	N	Y
Integrity: Connection Oriented With Recovery	N	N	N	Y	Y
Integrity: Connection Oriented Without Recovery	N	N	Y	Y	Y
Integrity: Selective Field and Connection Oriented	N	N	N	N	Y
Integrity: Connectionless	N	N	Y	Y	Y
Integrity: Selective Field Connectionless	N	N	N	N	Y
Non-repudiation: Origin or Delivery	N	N	N	N	Y

Key

* No services are allocated to OSI layer 5; layer 6, the presentation layer, contains a number of security facilities which support the provision of security services by the application layer.

** The services allocated to OSI layer 7, the application layer, may be provided by the application process itself.

N = Service not allocated to layer.

Y = Service allocated and should be provided for in layer protocol.

NOTE: This table needs to be revised to reflect pending changes to ISO 7498-2. This revision will be accomplished when the source information is finalized and analyzed.

Figure 3-1. Security Services Allocated to OSI Layers*

3.5.2.2.3 System Management Services

A security management application process should be used for establishing a security association for interactive communications among end systems. Implementation of communications applications (e.g., X.400 electronic mail, X.500 directory services, file transfer) and communications protocols will occur as untrusted applications within the end system software security architecture. Security protection for these untrusted applications should be provided by the establishment of a security association for an interactive communications dialog. A SMAP should be used to establish a security association.

End systems that support multiple information domains must also provide independent security management for each of the information domains. The security policy rules for both end system security management and information domain security management must be part of the end system security management information base. The relationship between the SMIB and a SMAP is described in Volume 6. The SMAP must be capable of responding to an end user application request for a specific security mechanism or be able to adopt a suitable one based on the information domain or end system security policies contained in the SMIB.

To allow for effective distribution of security management across many end system platforms, standardization of security management functions, data structures, and protocols is recommended. Specific areas for security management standardization are identified in Volume 6.

3.5.3 Guidance for Architectural Components Other Than End Systems or Relay Systems

This section provides a brief overview of the guidance in Volume 6 for the architectural components of local subscriber environment, local communications system, and communications network.

3.5.3.1 Local Communications System Guidance

Security services are generally not required of implementations in the LCS unless the LCS is only used for communications among end systems in the same LSE. Even if this condition is satisfied, care must be taken that implementations in the LCS do not interfere with requirement added at a later date for communications with end systems in other LSEs. Nonetheless, should implementations of security mechanisms in the LCS be desirable, use of the same approaches (protocols and security management applications) as described for the end system network services will apply.

3.5.3.2 Communications Network Guidance

Because of the use of common carriers for transmitting information, the CNs are expected frequently not to be under the control of the DoD, and perhaps not under the control of a DoD organization with a comparable or otherwise suitable DoD information domain security policy. Therefore, allocating security services other than availability to the CNs is not recommended. In addition to the general need for communications resource availability, this may also provide for protection against "denial of service" to specific applications.

APPENDIX A

REFERENCES

Note: References appearing in this section represent documents used in preparation of the TAFIM, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

1. Air Force Communications and Computer Systems Integration Guide, U.S. Air Force Technology Integration Center, Version 3, April 1991.

2. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. VI, Integrated Systems Control April 1987.

3. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. VII, Software Architecture, December 1990.

4. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. I, Overview, July 1990.

5. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. IV, Local Information Transfer, September 1987.

6. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. V, Long Haul Information Transfer, April 1987.

7. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. II, Deployable Architecture, September 1989.

8. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. III, Data Management Architecture, April 1990.

9. Air Force Planning & Architecture Guidance, Part II Architecture & Implementation Guide, U.S. Air Force, March 1990.

10. American National Standard for Information Systems - Dictionary for Information Systems, ANSI X3.172-1990, American National Standards Institute, July 1990.

11. Application Portability Profile, National Institute of Standards and Technology, April 1991, Army Tactical Command and Control Information System (ATCCIS), Working Paper 30, Applicability of the Architecture, January 1990.

12. Army Tactical Command and Control Information System (ATCCIS), Working Paper 22, Architectural Concepts, September 1987.

13. Army Tactical Command and Control Information System (ATCCIS), Working Paper 24, Architectural Definition, September 1990.

14. Army Tactical Command and Control Information System (ATCCIS), Working Paper 34, ATCCIS Communications, January 1990.

15. Army Tactical Command and Control Information System (ATCCIS), Working Paper 11, Functional Requirements Derived From Key Tasks, January 1990.

16. Army Tactical Command and Control Information System (ATCCIS), Working Paper 7N, Standardization of Data for Interoperability, September 1990.

17. Army Tactical Command and Control Information System (ATCCIS), Working Paper 25, Technical Standards for Command and Control Information Systems (CCISs), Edition 3, January 1992.

18. Army Command and Control Information Systems Commonalty with the Information Systems Architecture, Draft, U.S. Army Information Systems Engineering Command, April 1992.

19. Army Information Architecture, Department of the Army, Pamphlet 25-1, 20 August 1991.

20. Army Information Mission Area Information System Architecture Security, Draft, U.S. Army Information Systems Engineering Command, September 1991.

21. Army Information Systems Architecture Circa 1997, U.S. Army Information Systems Engineering Command, August 1989.

22. Army Information Systems Architecture (ISA), briefing, June 1992. Army Information Systems Architecture '97, Strategic Implementation Plan, U.S. Army Information Systems Engineering Command, April 1992.

23. Army Information Systems Architecture, Strawman Version, U.S. Army ISC, 30 March 1992.

24. Army Information Systems Architecture, Vol. II, Strategic and Sustaining Base Architecture, U.S. Army ISC, December 1991.

25. Army Information Systems Architecture, Vol. III, Technology and Standards, U.S. Army ISC, December 1991.

26. Army, Integrated Architecture Volume Mid-Range Technical Architecture, U.S. Army Information Systems Engineering Command, July 1991.

27. Army, Multimedia for the Information Systems Architecture, Coordination Draft, U.S. Army Information Systems Engineering Command, April 1992.

28. Atkinson, Malcolm et al., December 1989, "The Object-Oriented Database System Manifesto," *Proc. DOOD 1989.*

29. Berson, Alex, 1992, *Client/Server Architecture*, New York: McGraw-Hill, Inc.

30. Boeing Enterprise Network, Vol. I, Vision and Architecture, Boeing Co., November 1989.

31. Boeing Enterprise Network, Vol. II, General Guidelines and Principles for Transition to BNA Phase 2, Boeing Co., November 1989.

32. CALS Architecture Study, Vol. II, The Joint CALS Management Office, 30 June 1991.

33. CALS Architecture Study, Vol. I: Report, The Joint CALS Management Office, 30 June 1991.

34. CCIS, Command and Control Information System (CCIS) Architecture, MITRE, February 1990. CCIS, Generic and Target Architecture for Command and Control Information Systems, IDA Paper P-2490, September 1991.

35. CCIS, Survey of Technical Standards for Command and Control Information Systems, IDA Paper P-2457, September 1991.

36. CIM Human Computer Interface Style Guide, Version 1.0, February 1992.

37. CIM Review of Software Architectures, MITRE Corp., February 1992.

38. CIM Software Architecture Framework (Draft), MITRE Corp., April 1992.

39. CIM Software Development Framework, Draft, Center for Information Management, May 1992.

40. CIM Technical Reference Model for Information Management, Version 1.2, Center for Information Management, May 1992.

41. Copernicus Architecture, Implementation Plan for Phase II, Space and Naval Warfare Systems Command, December 1991.

42. Copernicus Architecture, Phase I: Requirements Definition, U.S. Navy, August 1991.

43. Copernicus Architecture, Phase I: Requirements Definition, Space and Naval Warfare Systems Command, December 1991.

44. Counter Narcotics, Information Protection Architecture, draft, Office of National Drug Control Policy, November 1991.

45. C4I For The Warrior Interoperability Tiger Team Final Report, Joint Staff, May 1992.

46. Defense Information Systems Agency, Client Server Migration Guidance for the Mission Support Area, Version 2.0, September 1993.

47. Defense Information Systems Agency, Defense Information System Network (DISN), A Goal Integrated Communications Architecture and Transition Strategy, Interim Report, April 1992.

48. Defense Information Systems Agency, Defense Information System Network (DISN), Final Report, September 1990.

49. Defense Logistics Agency (DLA), DoD Open Systems Life Cycle Management Concept for Corporate Information Management, October 1991.

50. Defense Logistics Agency (DLA), Office of Information Systems and Technology, Open Network Systems Implementation and Management, Version 1.1, September 1991.

51. Defense Logistics Agency (DLA), Office of Information Systems and Technology, Information Resources Management Environment Vision and Prescription, Version 1.1, April 1991.

52. Defense Logistics Agency (DLA), Systems Software Blueprint, DSAC System Software, June 1986.

53. Defense Logistics Agency (DLA), The DLA Open Systems Architecture for Information Systems, DSAC, December 1988.

54. Department of Defense (DoD) Command, Control, Communications, Computers, and Intelligence (C4I) for the Warrior Directive, draft, April 1992.

55. Department of Defense (DoD) Goal Security Architecture (DGSA) Executive Summary, draft, August 1993.

56. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) A Framework for Evolution of the Department of Defense Intelligence Information System (DoDIIS), Defense Intelligence Agency, July 1991.

57. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Client-Server Environment (CSE) Specification, The Engineering Review Board of the DoDIIS Management Board of the Defense Intelligence Agency, June 1991.

58. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Reference Model for the 1990s, DoDIIS Management Board, October 1991.

59. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Standards Document, The MITRE Corporation, October 1991.

60. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Style Guide, DoDIIS Management Board, October 1991.

61. Department of Defense (DoD) Software Technology Strategy, draft, Director of Defense Research and Engineering, December 1991.

62. Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, Department of Defense, December 1985.

63. DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Capacity Management, June 1989.

64. DSAC Office of Architectural Integration, Strategic Architectural Objective, DLA Communications, June 1989.

65. DSAC Office of Architectural Integration, Strategic Architectural Objective, Corporate Data System Application Design Objectives for Departmental and Personal Platforms, August 1990.

66. DSAC Office of Architectural Integration, Strategic Architectural Objective, Data Center Automation, June 1989.

67. DSAC Office of Architectural Integration, Strategic Architectural Objective, Operating Systems, January 1990.

68. DSAC Office of Architectural Integration, Strategic Architectural Objective, DLA Communications Configuration, June 1990.

69. DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Information Management, June 1989.

70. DSAC Office of Architectural Integration, Strategic Architectural Objective, Development Support, June 1989.

71. DSAC Office of Architectural Integration, Strategic Architectural Objective, Database Management System July 1990.

72. DSAC Office of Architectural Integration, Strategic Architectural Objective, Data Administration, January 1990.

73. Faulkner, March 1993, "Data Base Management Systems," *Faulkner Technical Report.*

74. FIPS Publication 11-3, Guideline: American National Dictionary for Information System, National Institute of Standards and Technology, February 1991.

75. FIPS Publication 146-2, Profiles for Open Systems Internetworking Technologies (POSIT), National Institute of Standards and Technology, 1996.

76. Functional Process Improvement, DoD 8020.1-M, draft, April 1992.

77. Global Transportation Network, C4S Technical Standards, U.S. Transportation Command, 15 April 1992.

78. Information Technology Portfolio Matrix, draft, June 1992.

79. International Organization for Standardization (ISO), 1984, Information Processing Systems, *Open Systems Interconnection Reference Model: Basic Reference Model*, ISO 7498.

80. International Organization for Standardization (ISO), 1989, Information Processing Systems, *Open Systems Interconnection Reference Model, Part 4: Management Framework*, ISO 7498.

81. JCS, Command Center System Architecture and TA/CE Guidance, FY 92-97, Volume I, JCS, September 1991

82. JCS, Joint Staff Automation for the Nineties (JSAN), Grumman Data Systems Corporation, March 1991.

83. MAGTF, Interoperability Requirements Concepts (MIRC), USMC, 4 May 1990.

84. Marine Corps Tactical Communications Architecture (MCTCA), DON, HQMC, 30 July 1990.

85. Nation Photographic Interpretation Center (NPIC) Information System Volume I, Architecture Definition, NPIC, December 1991.

86. Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture, International Organization for Standardization.

87. POSIX, Draft Guide to the POSIX Open Systems Environment, P1003.0/D15, draft, June 1992.

88. Software Technology for Adaptable, Reliable Systems (STARS), Updated System Specification (SSS), September 1990.

89. Standards-Based Architecture Planning Guide, Draft Version 1.2, DMR Group, Inc., 24 April 1992.

90. Strategies for Open Systems, Stage Four, Standards-Based Architectures, DMR Group, 1991.

91. Target Architecture and Implementation Strategy for the Joint MLS Technology Insertion Program, MTR-91W00134, The MITRE Corporation, September 1991.

92. UNIX, 1992 Road Map for System V and Related Technologies, UNIX International, February 1992.

93. UNIX, 1992 UNIX System V Release 4 Product Catalog, UNIX International, Spring 1992

APPENDIX B

ACRONYMS AND DEFINITIONS

ACRONYMS

AIS	Automated Information System
ANSI	American National Standards Institute
API	Application Program Interface
APP	Application Portability Profile
BBS	Bulletin Board System
C3I	Command, Control, Communications, and Intelligence
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CASE	Computer-Aided Software Engineering (See ISEE)
CIM	Corporate Information Management
CJCS	Chairman of the Joint Chiefs of Staff
CN	Communications Network
CORBA	Common Object Request Broker Architecture
COTS	Commercial-Off-the-Shelf
CODASYL	Conference on Data Systems Languages
DBMS	Database Management System
DD/DS	Data Dictionary/Directory System
DGSA	Defense Goal Security Architecture
DISA	Defense Information Systems Agency
DMS	Defense Message System
DoD	Department of Defense
DoDD	Department of Defense Directive
DISN	Defense Information Systems Network
E-mail	Electronic Mail
EEI	External Environment Interface
ES	End System
FIPS	Federal Information Processing Standard
FTAM	File Transfer, Access, and Management

FTP	File Transfer Protocol
GOSIP	Government Open System Interconnection Profile
GSS	General Security Service
GUI	Graphical User Interface
IBM	International Business Machines
IETF	Internet Engineering Task Force
IM	Information Management
IRDS	Information Resource Dictionary System
ISAM	Indexed Sequential Access Method
ISO	International Organization for Standardization
IT	Information Technology
ITSI BBS	Information Technology Standards Information Bulletin Board System
JIEO	Joint Interoperability and Engineering Organization
LAN	Local Area Network
LCS	Local Communications System
LSE	Local Subscriber Environment
MAN	Metropolitan Area Network
MHS	Message Handling System
MIL-STD	Military Standard
MOP	Memorandum of Policy
NCSC	National Computer Security Center
NIST	National Institute of Standards and Technology
OLE	Object Linking and Embedding
OODBMS	Object-Oriented Database Management System
ORB	Object Request Broker
OSI	Open System Interconnection
OMG	Object Management Group
POSIT	Profiles for Open Systems Internetworking Technologies
POSIX	Portable Operating System Interface (for Computer Environments)
RDBMS	Relational Database Management System
RS	Relay System

SAMP	Security Association Management Protocol
SMAP	Security Management Application Process
SMIB	Security Management Information Base
SMTP	Simple Mail Transfer Protocol
SNA	System Network Architecture
SQL	Structured Query Language
SWG	Special Working Group
TAFIM	Technical Architecture Framework for Information Management
TCP/IP	Transmission Control Protocol/Internet Protocol
TRM	Technical Reference Model
WAN	Wide Area Network
WWW	World Wide Web

DEFINITIONS

Application–The use of capabilities (services and facilities) provided by an information system specific to the satisfaction of a set of user requirements. [P1003.0/D15]

Application Platform–The collection of hardware and software components that provide the services used by support and mission-specific software applications.

Application Portability Profile (**APP**)–The structure that integrates Federal, national, international, and other specifications to provide the functionality necessary to accommodate the broad range of federal information technology requirements. [APP]

Application Program Interface (**API**)–(1) The interface, or set of functions, between the application software and the application platform. [APP] (2) The means by which an application designer enters and retrieves information.

Architecture–Architecture has various meanings depending upon its contextual usage. (1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE STD 610.12] (2) Organizational structure of a system or component. [IEEE STD 610.12]

Architecture, Database–The logical view of the data models, data standards, and data structure. It includes a definition of the physical databases for the information system, their performance requirements, and their geographical distribution. Ref DoD 8020.1-M, Appendix J

Architecture Target–Depicts the configuration of the target open information system. Ref DoD 8020.1-M

Architecture, Infrastructure–Identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. [DoD 8020.1-M, Appendix J specifically paragraph 5(14)(c), Table J-2]

Architectural Structure–Provides the conceptual foundation of the basic architectural design concepts, the layers of the technical architecture, the services provided at each layer, the relationships between the layers, and the rules for how the layers are interconnected.

Automated Information System (**AIS**)–Computer hardware, computer software, telecommunications, information technology, personnel, and other resources that collect, record, process, store, communicate, retrieve, and display information. An AIS can include computer software only, computer hardware only, or a combination of the above. [DoDD 8000.1]

Baseline–A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and that can be changed only through formal change control procedures or a type of procedure such as configuration management. [IEEE STD 610.12]

Commercial-Off-The-Shelf (COTS)–Refers to an item of hardware or software that has been produced by a contractor and is available for general purchase. Such items are at the unit level or higher. Such items must have been sold and delivered to government or commercial customers, must have passed customer's acceptance testing, be operating under customer's control, and within the user environment. Further, such items must have meaningful reliability, maintainability, and logistics historical data.

Communications Link–The cables, wires, or paths that the electrical, optical, or radio wave signals traverse. [TA]

Communications Network–A set of products, concepts, and services, that enable the connection of computer systems for the purpose of transmitting data and other forms (e.g., voice and video) between the systems.

Communications Node–A node that is either internal to the communications network (e.g., routers, bridges, or repeaters) or located between the end device and the communications network to operate as a gateway. [TA]

Communications Services–A service of the Support Application entity of the Technical Reference Model that provides the capability to compose, edit, send, receive, forward, and manage electronic and voice messages and real time information exchange services in support of interpersonal conferencing. [TA]

Communications System–A set of assets (transmission media, switching nodes, interfaces, and control devices) that will establish linkage between users and devices.

Configuration Management–A discipline applying technical and administrative direction and surveillance to: (a) identify and document the functional and physical characteristics of a configuration item, (b) control changes to those characteristics, and (c) record and report changes to processing and implementation status.

Connectivity Service–A service area of the External Environment entity of the Technical Reference Model that provides end-to-end connectivity for communications through three transport levels (global, regional, and local). It provides general and applications-specific services to platform end devices. [TA]

Database Utility Service–A Service of the Support Application Entity of the Technical Reference Model that provides the capability to retrieve, organize, and manipulate data extracted from a database. [TA]

Data Dictionary–A specialized type of database containing metadata, which is managed by a data dictionary system; a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases; an application of data dictionary systems. [DoDD 8320.1]

Data Element–A basic unit of information having a meaning and that may have subcategories (data items) of distinct units and values. [DoDD 8320.1]

Data Interchange Service–A service of the Platform entity of the Technical Reference Model that provides specialized support for the interchange of data between applications on the same or different platforms. [TA]

Data Management Service–A service of the Platform entity of the Technical Reference Model that provides support for the management, storage, access, and manipulation of data in a database. [TA]

Directory Service–A service of the External Environment entity of the Technical Reference Model that provides locator services that are restricted to finding the location of a service, location of data, or translation of a common name into a network specific address. It is analogous to telephone books and supports distributed directory implementations. [TA]

Distributed Database–(1) A database that is not stored in a central location but is dispersed over a network of interconnected computers. (2) A database under the overall control of a central database management system but whose storage devices are not all attached to the same processor. (3) A database that is physically located in two or more distinct locations. [FIPS PUB 11-3]

Enterprise–The highest level in an organization – includes all missions and functions. [TA]

Enterprise Model–A high-level model of an organization's mission, function, and information architecture. The model consists of a function model and a data model.

External Environment Interface (EEI)–The interface that supports information transfer between the application platform and the external environment. [APP]

Functional Architecture–The framework for developing applications and defining their interrelationships in support of an organization's information architecture. It identifies the major functions or processes an organization performs and their operational interrelationships. [DoD 5000.11-M]

Functional Area–A range of subject matter grouped under a single heading because of its similarity in use or genesis. [DoDD 8320.1]

Function–Appropriate or assigned duties, responsibilities, missions, tasks, powers, or duties of an individual, office, or organization. A functional area is generally the responsibility of a PSA (e.g., personnel) and can be composed of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviews). Ref Joint Pub 1-02, DoDD 8000.1, and DoD 8020-1M.

Functional Activity Program Manager (FAPM)–FAPMs are designated by PSAs and are accountable for executing the functional management process. Supported by functional representatives from the DoD Components, FAPMs develop functional architectures and strategic plans, and establish the process, data, and information system baselines to support functional activities within the functional area Ref DoD 8020.1-M Ch 1 B(2).

Functional Data Administrator (FDAd)–OSD PSAs exercise or, designate functional data administrators to perform data administrator responsibilities to support execution of the functional management process, and to function within the scope of their overall assigned responsibilities. Ref DoDD 8320.1 and DoD 8020.1-M, Appendix A.

Functional Economic Analysis (FEA)–A structured proposal that serves as the principal part of a decision package for enterprise (individual, office, organization - see function) leadership. It includes an analysis of functional process needs or problems; proposed solutions, assumptions, and constraints; alternatives; life-cycle costs; benefits and/or cost analysis; and investment risk analysis. It is consistent with, and amplifies, existing DoD economic analysis policy. Ref DoDI 7041.3, DoDD 8000.1, and DoD 8020.1-M, Appendix H.

Hardware–(1) Physical equipment, as opposed to programs, procedures, rules, and associated documentation. (2) Contrast with software. [FIPS PUB 11-3]

Information–Any communication or representation of knowledge such as facts, data, or opinions, in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual forms. [OMB CIRC A-130]

Information Domain–A set of commonly and unambiguously labeled information objects with a common security policy that defines the protections to be afforded the objects by authorized users and information management systems. [DISSP]

Information Management(IM)–The creation, use, sharing, and disposition of information as a resource critical to the effective and efficient operation of functional activities. The structuring of functional processes to produce and control the use of data and information within functional activities, information systems, and computing and communications infrastructures. [DoDD 8000.1]

Information Resources Management (IRM)–The planning, budgeting, organizing, directing, training, promoting, controlling, and management activities associated with the burden (cost), collection, creation, use, and dissemination of information by Agencies and includes the management of information and related resources, such as federal information processing (FIP) resources. Ref PL No 99-591, DoDD 8000.1.

Information Technology (IT)–The technology included in hardware and software used for Government information, regardless of the technology involved, whether computers, communications, micro graphics, or others. Ref OMB Circular A-130 and DoDD 8000.1.

Infrastructure–Infrastructure is used with different contextual meanings. Infrastructure most generally relates to and has a hardware orientation but note that it is frequently more comprehensive and includes software and communications. Collectively, the structure must meet the performance requirements of and capacity for data and application requirements. Again note that just citing standards for designing an architecture or infrastructure does not include functional and mission area requirements for performance. Performance requirement metrics must be an inherent part of an overall infrastructure to provide performance interoperability and compatibility. It identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. Ref DoD 8020.1-M

Interoperability–The ability of systems to exchange useful data and information.

Legacy Environments–Legacy environments could be called legacy architectures or infrastructures and as a minimum consist of a hardware platform and an operating system. Legacy environments are identified for phase-out, upgrade, or replacement. All data and applications software that operate in a legacy environment must be categorized for phase-out, upgrade, or replacement.

Legacy Systems–Systems that are candidates for phase-out, upgrade, or replacement. Generally legacy systems are in this category because they do not comply with data standards or other standards. Legacy system workloads must be converted, transitioned, or phased out (eliminated). Such systems may or may not operate in a legacy environment.

Life Cycle–The period of time that begins when a system is conceived and ends when the system is no longer available for use. [IEEE STD 610.12]. AIS life cycle is defined within the context of life-cycle management in various DoD publications. It generally refers to the usable system life.

Local Area Network (LAN)–A data network, located on a user's premises, within a limited geographic region. Communication within a local area network is not subject to external regulation; however, communication across the network boundary may be subject to some form of regulation. [FIPS PUB 11-3].

Migration Systems–An existing AIS, or a planned and approved AIS, that has been officially designated to support common processes for a functional activity applicable to use DoD-wide or DoD Component-wide. Systems in this category, even though fully deployed and operational, have been determined to accommodate a continuing and foreseeable future requirement and, consequently, have been identified for transitioning to a new environment or infrastructure. A migration system may need to undergo transition to the standard technical environment and standard data definitions being established through the Defense IM Program, and must "migrate" toward that standard. In that process it must become compliant with the Reference Model and the Standards Profile. A system in this category may require detailed analysis that involves a total redesign, reprogramming, testing, and implementation because of a new environment and how the "users" have changed their work methods and processes. The detailed analysis may identify the difference between the "as is" and the "to be" system. [DoD 8020.1-M].

Multimedia Service–A service of the TRM that provides the capability to manipulate and manage information products consisting of text, graphics, images, video, and audio. [TA]

Open Specifications–Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards. [P1003.0/D15]

Open System–A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [P1003.0/D15]

Open Systems Environment (OSE)–The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of applications, data, or people, as specified by information technology standards and profiles. [P1003.0/D15]

Operating System Service–A core service of the Platform entity of the Technical Reference Model that is needed to operate and administer the application platform and provide an interface between the application software and the platform (e.g., file management, input/output, print spoolers). [TA]

Platform–The entity of the Technical Reference Model that provides common processing and communication services that are provided by a combination of hardware and software and are required by users, mission area applications, and support applications. [TA]

Portability–(1) The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE STD 610.12] (2) A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment. [IEEE TUTOR] (3) The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TA]

Process Model–Provides a framework for identifying, defining, and organizing the functional strategies, functional rules, and processes needed to manage and support the way an organization does or wants to do business--provides a graphical and textual framework for organizing the data and processes into manageable groups to facilitate their shared use and control throughout the organization. [DoD 5000.11-M]

Profile–A set of one or more base standards, and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [P1003.0/D15]

Profiling–Selecting standards for a particular application. [P1003.0/D15]

Scalability–The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII]. The capability to grow to accommodate increased work loads.

Seamless Interface–Ability of facilities to call one another or exchange data with one another in a direct manner. Integration of the user interface that allows a user to access one facility through another without any noticeable change in user interface conventions. [DSAC SYS IM]

Stovepipe System–A system, often dedicated or proprietary, that operates independently of other systems. The stovepipe system often has unique, nonstandard characteristics.

System–People, machines, and methods organized to accomplish a set of specific functions. [FIPS PUB 11-3]

System Management Service–A service of the Platform entity of the TRM that provides for the administration of the overall information system. These services include the management of information, processors, networks, configurations, accounting, and performance. [TA]

Technical Reference Model (TRM)–The document that identifies a target framework and profile of standards for the DoD computing and communications infrastructure. [TRM]

User–(1) Any person, organization, or functional unit that uses the services of an information processing system. (2) In a conceptual schema language, any person or any thing that may issue or receive commands and messages to or from the information system. [FIPS PUB 11-3]

User Interface Service–A service of the Platform entity of the Technical Reference Model that supports direct human-machine interaction by controlling the environment in which users interact with applications. [TA]

REFERENCES FOR DEFINITIONS

[AF 700-50,V] Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. V, Long Haul Information Transfer, April 1987.

[APP] NIST Special Report, APP: The US. Governments Open System Environment Profile, Draft, January 1991.

[ARMY 25-1] Army Information Architecture, Department of Army, Pamphlet 25-1, 20 August 1991.

[BOEING] Boeing Enterprise Network, Vol. I, Vision and Architecture, Boeing Co., November 1989.

[DATE] Date, C. J. with Colin J. White, A Guide to DB2, Third Edition, Addison-Wesley Publishing Co., Reading, MA, 1989.

[CIM] DRAFT CIM Architecture Framework, November 1991.

[DISSP] Defense Information Systems Security Program

[DoD 5000.11-M] Department of Defense, DoD Data Administration Procedures, Department of Defense Manual 5000. 11-M Draft, 30 June 1991.

[DoD 8320.1-M] Department of Defense Data Administration Working Group, DoD Data Administration Procedures Manual, DoD Manual 8320. 1-M Draft, 25 October 1991.

[DoDD 5000.29] Department of Defense, Management of Computer Resources in Major Defense Systems, Department of Defense Directive 5000.29, 26 April 1976.

[DoDD 8000.1] Defense Information Management (IM) Program, Department of Defense Directive 8000.1, 27 October 1992.

[DoDD 8320.1] Department of Defense Data Administration, Department of Defense Directive 8320.1, 26 September 1991.

[DSAC SYS IM] DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Information Management, June 1989.

[ELMAGARMID] Elmagarmid, Ahmed K., and Calton Pu, "Guest Editors' Introduction to the Special Issue of Heterogeneous Databases," ACM Computing Surveys, Volume 22, Number 3, September 1990.

[FIPS PUB 11-3] FIPS Publication 11-3, Guideline American National Dictionary for Information System, National Institute of Standards and Technology, February 1991.

[FRAMEWORK] Software Development Framework, Draft, 15 May 1992.

[HCI] DISA, Human Computer Interface Style Guide, Version 1.0, 12 February 1992.

[IEEE STD 610.12] Institute of Electrical and Electronics Engineers, Inc., IEEE Standard Glossary of Software Engineering Terminology, IEEE STD 610.12-1990, 10 December 1990.

[IEEE TUTOR] Standards, Guidelines, and Examples on System and Software Requirements Engineering, Merline Dorman and Richard Thayer, editors, IEEE Computer Society Press Tutorial, 1990.

[KORTH] Korth, Henry F. and Abraham Silberschatz, Database System Concepts, McGraw-Hill Book Company, New York, 1986.

[MCC] McClure, Carma, The Three R's of Software Automation," Prentice Hall, 1992.

[OMB CIRC A-130] Office of Management and Budget Circular A-130, "Management of Information Resources," 1985.

[P1003.0/D15] Technical Committee on Operating Systems and Application Environments of the IEEE Computer Society, "Standards Project Draft Guide to the POSIX Open Systems Environment," June 1992.

[STALLINGS] Stallings, William, Business Data Communications, MacMillan Publishing Company, New York, 1990.

[TA] DoD TAFIM for Information Management (Draft).

[TRANSCOM] Global Transportation Network, C4S Technical Standards, US. Transportation Command, 15 April 1992.

[TRM] CIM Technical Reference Model for Information Management, Version 1.2, Center for Information Management, May 1992

[USAICII] Army Information Systems Architecture, Vol. II, Strategic and Sustaining Base Architecture, US. Army ISC, December 1991.

[WEBSTER] Webster's II New Riverside University Dictionary, The Riverside Publishing Company, MA, 1988.

APPENDIX C

OPEN SYSTEMS

A critical objective of the DoD Information Management initiative is the implementation of a computing and communications infrastructure that supports portability, interoperability, and scalability. To achieve this objective the DoD must develop and use open systems. The following definitions apply to open systems:

- OPEN SYSTEM: A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered hardware and applications software to:

    - Interoperate with other applications on local and remote systems

    - Be ported with minimal changes across a wide range of systems

    - Interact with users in a style that facilitates user portability

    - Enable users to increase processing power as their functional needs grow, without the need to re-write applications (i.e., scalability)

- OPEN SPECIFICATION: Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards.

- OPEN SYSTEM ENVIRONMENT (OSE): The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability and scalability, or for portability of applications, data, or people, as specified by information technology standards and profiles.

- OPEN SYSTEM ARCHITECTURE: The framework describing the entities (e.g., components, services) and their interrelationships in an open system.

Open systems environments and architectures are intended to help achieve portability, interoperability, scalability and cost effectiveness of systems. These attributes facilitate technology insertion and rapid system evolution to respond to changing functional practices – functional and technical managers will have the capability to selectively preserve or reconfigure parts of the infrastructure based on functional needs.

Open systems are modular, enabling users to define, acquire, and add to systems that are supplied by a variety of vendors in an open, competitive market. An open system supports the interoperability of hardware, software, and communications products developed by different suppliers at different times.

DoD information systems will incrementally evolve to converge towards open system architecture guidelines and standards while accommodating existing baselines and transition environments. Implementation guidelines will be provided for engineering and economic analysis of options and opportunities to evolve baselines to target architectures. They will support the decision making process to select the best overall targets and transition paths.

APPENDIX D

PROPOSING CHANGES TO TAFIM VOLUMES

D.1 INTRODUCTION

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section D.2. Guidance on the use of the format is provided in Section D.3.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown in Section D.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail: **TAFIM**
Booz Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/671-7937**; indicate "TAFIM" on cover sheet.

D.2 TAFIM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

D.3 FORMAT GUIDANCE

The format in Section D.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**D.2 a**) of the format would identify that individual. The information in the **Document Identification** part of the format (**D.2 b**) is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts (**D.2 c, D.2 d, or D.2 n**) of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: "The working baseline will only be available to the TAFIM project staff." The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The D.2 c (5), D.2 d (5), or D.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in D.2 c (6), D.2 d (6), or D.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

This page intentionally left blank.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 4: DoD Standards-Based Architecture Planning Guide

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 4 of the TAFIM, contains no substantive changes from Volume 4 of Version 2.0. Minor modifications have been made to acknowledge the evolving policies noted above. Substantive revisions to reflect these policy changes fully will be made in the next edition.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group (AMWG) will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.
- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.
- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.
- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. The Defense Information Systems Agency (DISA) is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

Preface

A key element of the United States (U.S.) Department of Defense'S (DoD) Corporate Information Management (CIM) initiatives for the 1990s is the implementation of a computing and communications infrastructure that will support portability, scalability, and interoperability of applications.

Deputy Secretary of Defense William J. Perry's policy memorandum of 13 October 1993 entitled "Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement" reaffirms CIM principles and calls for all DoD components to begin migration from legacy to target systems in such a way "that migrate the system toward an open system environment and a standards-based architecture defined by the DoD Technical Architecture Framework for Information Management" (TAFIM).

In support of this goal, the ***DoD Standards-Based Architecture Planning Guide*** (the SBA Guide) has become Volume 4 of the TAFIM, which defines a common framework and profile of standards for the computing and communications infrastructure. The methodology prescribed in the SBA Guide provides a way of mapping the technology architecture, which is the primary focus of Volumes 1, 2, and 3 of TAFIM, to the three other views of an integrated architecture: work, data or information, and applications.

This version of the SBA Guide is an update of an earlier version that was written from October 1991 through April 1992 under contract #DCA100-91-C-0166. It presents a process for developing a standards-based architecture within the Department of Defense. At the time of this update, two major architecture engagements have been completed based on the use of the planning approach described in the earlier version of the SBA Guide. The goal of the updated SBA Guide is to incorporate recommended changes that effectively echo the lessons learned in the course of these two engagements.

The first major implementations of the SBA Guide were intended to test the methodology in a "small" enterprise (*Office of the Secretary of Defense, Office Automation Standards-Based Architecture*) and in a large-scale enterprise implementation (*Standards-Based Architecture for the U.S. Marine Corps).* The documents created as a result of these initiatives currently constitute the best reference source for the expected output from such an effort.

The planning process itself specifically addresses the Information Technology Policy Board (ITPB) Task 91-01 policy proposal approved 10 April 1991, which states:

> Develop a DoD standards-based open systems information systems architecture development methodology and establish a DoD implementation strategy.

The earlier version of the document was based on DMR Group, Inc.'s *Standards-Based Architectures*, Vol. IV in the STRATEGIES FOR OPEN SYSTEMS research program. This document, and its underlying architecture development process, was unique in its:

- New approach to gaining functional management understanding of, support for, and involvement in the information systems architecture process
- Explicit determination of broad organizational information systems architecture principles
- Explicit approach to creating an architecture based on standards
- Express design to produce "vendor neutral" architectures
- Proven application across a wide range of organizational types
- Immediate availability.

The process described herein is specifically designed so that all target architectures derived through its use meets standards and incorporates the generalized guidance on open systems environments (OSE) found in National Institute of Standards and Technology (NIST) Special Publication 500-187, "Applications Portability Profile

(APP): The U.S. Government's Open Systems Environment Profile OSE/1 Version 2.0."

Corporate Information Management practices and policies are still evolving. As they do, this SBA Guide will also require changes in its diagrammatic representations, terminology, and policy discussions.

This page intentionally left blank.

An Executive Summary

This document was developed to assist users in the United States (U.S.) Department of Defense (DoD) in planning technology architectures based on standards-based platforms. It can be used within a functional unit or department within the DoD (e.g., the Marine Corps). The approach may also be usefully applied at a lower, or sub-department, level to provide a more detailed view of the architecture.

Target audience

Process facilitators constitute the primary audience of interest for whom this document was created. Experience tells us that this planning process is most successful when it is led by someone who can bring an impartial view to bear on the consensus-building process that is central to the success of the effort. An impartial and professional facilitator, experienced in the standards-based architecture (SBA) process, is essential in getting the process off the ground. The facilitator will keep the process on track when local, political, or technical perspectives threaten to get things moving in the wrong direction or risk derailing the process. This is said in recognition of the fact that many of those asked to participate in the process are likely to bring with them parochial views or hidden agendas that might not allow them to work effectively toward the common goal of developing a mission-specific architecture. The best way to address these issues is through reliance on a facilitator who can identify stumbling blocks and move the team around or over them.

The facilitator will have experience in facilitating workshop sessions with key knowledge workers to elicit required architectural content. The facilitator will also possess the ability to tailor the basic methodology as needed to satisfy the unique demands of the enterprise being modeled. Thus, for the facilitator, this SBA Guide becomes a sourcebook for customizing the specific methodology to meet the specific goals of the organization involved.

Other audiences will also be interested in this document. It can be used as a marketing tool to expose prospective participants or sponsors to the process and educate them about what the SBA planning process can achieve. It is also useful to those involved in the process to help them understand the importance of each step they are involved in and how one step serves as a basis for work to be done in successive steps of the process.

The SBA Guide can be used to "hand hold" those involved who may occasionally feel lost or overwhelmed by the task in which they are involved.

This document *is not* a detailed methodology describing all attributes of information modeling, application development, security architecture, or detailed technical implementation project planning, nor does it describe the methodology by which "business process redesign" is accomplished.

While this document discusses such subjects, it is not intended to provide the reader with a detailed understanding of those methodologies and techniques. Furthermore, it was not designed to develop a single monolithic DoD architecture for a single computer and communications solution that will fit all users across the Defense community.

Volume 4 of TAFIM

The SBA methodology that is described in this guide is based on four views of an integrated architecture: work organization, information, applications, and technology. Volumes 1, 2, and 3 of the Technical Architecture for Information Management (TAFIM) focus primarily on the technology architecture. The SBA methodology, which constitutes Volume 4 of the TAFIM, provides a way of mapping the three other views (work, information, and applications) to the technology architecture.

The SBA planning process is an especially important part of the TAFIM because it fleshes out the work, information, and application views of the architecture. It provides a mechanism for translating the functional, or business, needs of the enterprise into the information technology (IT)-based solutions that ultimately flow from implementation of the entire TAFIM process.

The planning process helps align IT with the business needs of the organization. The *DoD Standards-Based Architecture Planning Guide* describes how the overall process of planning for, and implementing, a standards-based architecture is conducted, highlighting some key considerations in the overall effort. Because of the velocity of change in technology, which seems to be increasing, this process may be amended, adopted, and modified to conform to existing IT planning approaches that may already exist in DoD functional areas. Most importantly, it outlines a simple but effective process users may follow to arrive at a technology architecture based on standards.

Reusable building blocks

The ultimate goal of such a process is to yield reusable building blocks that can be used in each additional DoD component as it launches its own SBA planning process for the first time. While this version of the SBA Guide is based on two completed implementations, two other implementations are already under way, with other additional projects expected to follow. Some of the output from the past implementations is beginning to be replicated in the next round. As the DoD develops more understanding of the similarities observed across the entire organization, it can begin to understand how entire business processes may be supported by architecture in an identical way across the various components.

As an example, "work process" may be seen to constitute a reusable building block of the larger enterprise. If the work process "Acquiring Personnel" becomes a standard work process across DoD departments, then the IT architecture that supports this business, or work, process can be borrowed from implementation plans already available rather than having to "reinvent the wheel!"

Figure 1 represents the standards-based architecture planning and implementation cycle outlined in this SBA Guide.

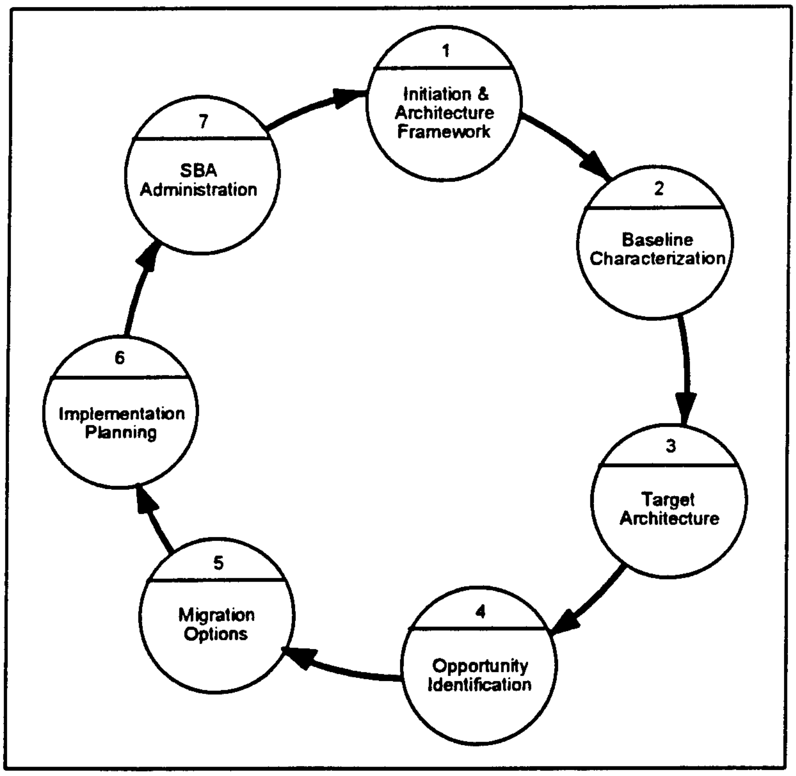
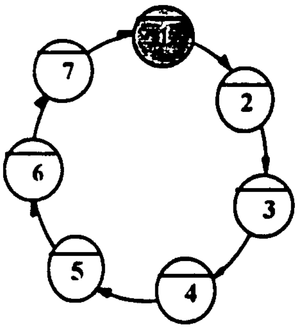

1
Initiation & Architecture Framework
2
Baseline Characterization
3
Target Architecture
4
Opportunity Identification
5
Migration Options
6
Implementation Planning
7
SBA Administration

Figure 1. The DoD Standards-Based Architecture (SBA) Planning Process

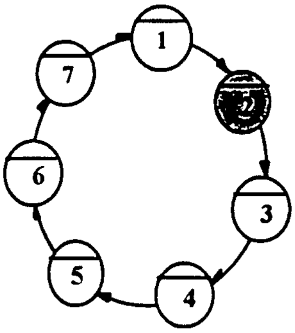
SBA process steps



The SBA planning process consists of seven distinct, but interdependent, phases. Each phase of the SBA process is intended to create specific deliverables which then guide the subsequent step(s). The phases and their deliverables are briefly outlined below:

1. **Initiation and architecture framework.** The methodology begins by properly initiating the process within the host organization. Once the process is properly sponsored and staffed for optimum effectiveness, it is possible to move on to the actual steps necessary to develop the architecture.

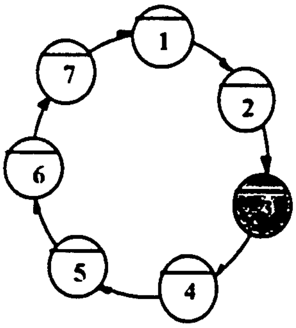
   This orientation phase involves reviewing (or in some cases developing) a set of strategic drivers for the organization. The business model is reviewed (or built) during this project phase to establish a strategic target operational model. Lastly, a set of architecture principles is developed, usually in workshops, to establish what are believed to be good architecture practices for the organization.



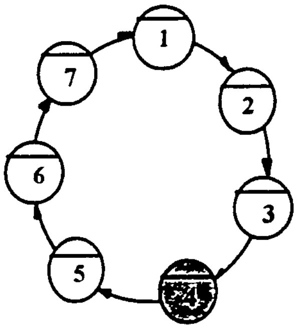
2 . **Baseline characterization.** This is a grounding phase to determine where an organization is currently situated architecturally. It is not an operational review or audit but more an assessment and characterization of the current environment. It is used to establish a baseline or starting point for architecture development. The architecture framework provides an effective means for organizing this review and presenting the current status.

The baseline characterization phase results in a picture of the existing architecture along four key dimensions, or views: work, information, applications, and technology. The term "characterization" is used because the data gathering and analysis are not exhaustive. It is not necessary, nor is it desirable, to expend the time and effort to document every detail of the current architecture. Only enough detail is gathered to allow informed decisions to be made with regard to the desired target architecture (described below).

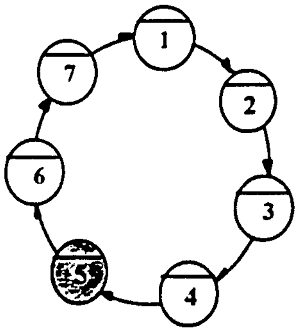
The current situation in each of the four views and their interrelationships will be characterized by completing a series of instruments, or templates. These templates are similar in content and style to the deliverables that will be used to define a target architecture. This will facilitate "gap analysis" for migration and implementation planning in future phases.



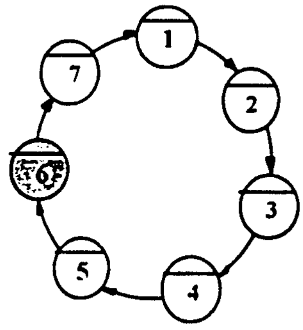
3 . **Target architecture.** This is the heart of the process, where the various views of the framework are modeled in terms of a desirable target architecture, usually 3 to 5 years in the future. The process consists of defining each set of architectural components and its key attributes. The components are then used to define desired relationships using affinity analysis. The result is an organized set of definitions and models from which drawings can be made to reflect the different views of the architecture.



4. **Opportunity identification.** This phase moves the architecture out of the conceptual world into one where the practical realities govern implementation. In this step, short-term opportunities are identified which, once implemented, can demonstrate the value of the architecture and provide immediate benefits to the organization. In addition, all projects that are necessary to achieve the target architecture are identified and fleshed out in some detail.



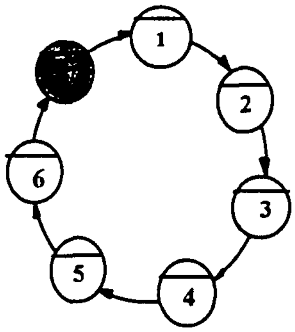
5. **Migration options.** This phase links the reality of the present with the desirability of the target architecture by establishing one or more plateaus representing practical migration stages. The same types of models, using the common framework, can be used to represent these evolutionary plans. All projects identified in the previous step are prioritized over time based on inter-project dependencies and cost/benefit analyses.



6. **Implementation planning.** This phase results in a detailed implementation plan for the first plateau of the migration effort. It constitutes the first wave of actionable projects that establish the groundwork for each successive plateau of the target architecture implementation. Plateau 1 projects are generally linked to the next stage in the migration plan. Responsibilities are established to ensure that they are carried out and that the migration plan is properly updated.

The outward manifestation of the architecture is also reflected in a set of standards and guidelines to be used by the organization in acquiring technology and developing applications. They can relate to any or all components in the models. Areas where standards are required most urgently can be identified for quick resolution and others assigned for later investigation.

The activity of identifying standards and guidelines for technology acquisition is informed by Volume 2 of TAFIM and by guidance provided by other Government-sponsored initiatives such as the Application Portability Profile (APP) developed by the National Institute of Standards and Technology (NIST).



7 . **SBA administration.** This phase is intended to keep the architecture alive and well by continuously improving it. This phase reflects the need to adjust architecture decisions in accordance with unforeseen changes in business directions or advances in technology or its availability. It should also be used to make adjustments based on experience and ensure that modifications in standards and supporting processes reflect a realistic approach. This review process can cause a reentry into the process at any point depending on the area to be adjusted or updated.

Essentially, this management activity ensures that the SBA planning process already is, or is soon to become, well integrated with the mainstream IT planning process within the organization. If it is treated as a special project, or in other ways is not fully institutionalized, the ability of the process to result in funded projects will ultimately suffer. The outcome of this step is a direct reflection of how successful the project initiation was in the first place. We cannot overemphasize the importance of properly positioning this process within the day-to-day operation. High-level sponsorship at the front end will contribute to success at the back end. This is true for a number of reasons that are discussed in Section 8.

Critical success factors

Experience has shown that there are lessons to be learned in how best to conduct architecture planning. The following represents a list of critical success factors that have been established:

Business driven

Wherever possible, use the architecture process to reinforce support of key operational and business drivers.

Participative process

Involve teams of architects, planners, and managers directly in the creation and review of deliverables. Establish corporate "buy-in."

Fast paced

Set schedules such that deliverables arrive within weeks, not months. Show early results.

Presumptive resolution

Do not get bogged down if facts or information are not available. Be presumptive, make the best guess, and document assumptions.

Architecture, not design

Avoid too much detail. Focus on architecture decisions and save some creative work for the designers to follow.

Minimum set

Do not set out to establish standards for everything in sight. Focus on those where key infrastructure is involved and leave the user departments to sort out the rest.

Key deliverables

It is more important to produce results that everyone can abide by than to follow specific processes or methods. Use the framework but be creative and experimental with methods using standard DoD tools and techniques.

Open, non-secretive

Do not hide the team away and stamp everything "confidential!" Invite participation and circulate drafts for review and discussion. Avoid alarming affected parties.

Ongoing process, not event

This is not intended to produce a shelf document and then allow everyone to get back to their former ways of making IT decisions. Creating ongoing processes for updating and reviewing are critical.

The SBA Guide is organized around the seven phases and associated critical success factors.

Overview of the DoD SBA Guide

This SBA Guide contains eight sections, each dealing with a specific topic:

Section 1
Introduction

Provides a context for this document and describes what the SBA planning process is and why it is important.

Section 2
Initiation and Architecture Framework

Describes Phase 1 of the process whereby an organization develops "architecture principles" and develops a common vision for the development of a standards-based technology architecture.

Section 3
Baseline Characterization

Outlines the overall process that is followed to conduct a high-level inventory of applications, platforms, and standards in place in the function.

Section 4
Target Architecture

Defines the steps and processes involved in developing a target architecture based on standards.

Section 5
Opportunity Identification

Illustrates how the Architecture Working Group (AWG) categorizes and identifies opportunities for exploiting the target architecture.

Section 6
Migration Options

Provides a framework for developing migration options to the new standards-based architecture.

Section 7
Implementation Planning

Defines how implementation project planning occurs and describes the steps by which the near- and mid-term benefits of the architecture are obtained.

Section 8
SBA Administration

Looks at the challenge of improving the new architecture over time to assure that incremental improvements are made on a continuous basis.

Appendices

These provide in-depth content and guidance in selected areas outlined by the individual sections.

This page intentionally left blank.

Table of Contents

List of Figures

Section One: Introduction

Section Two: Initiation and Architecture Framework

Section Three: Baseline Characterization

Section Four: Target Architecture

Section Five: Opportunity Identification

Section Six: Migration Options

Section Seven: Implementation Planning

Section Eight: SBA Administration

This page intentionally left blank.

Section One: Introduction

Table of Contents

Figures

Section description

As the *Foreword: An Executive Summary* stated, this document is not a formal methodology. It is a standards-based approach to standards. Why are standards so important to IT architecture? Simply put:

A new technology paradigm based on the concept of open network computing is emerging. It is driven by advances in technology and a combination of growing interdependence and heightened competition among functional organizations. Standards are "the glue" that enable users to interoperate seamlessly across applications, platforms, and organizations. Today's reality is that users are confronted with islands of automation—myriad and redundant computer systems that have been used to automate non-standard, and frequently inefficient, functional processes.

Standards-based environments are delivering important benefits to organizations in two main categories: reduced cost of IT and its management, and improved IT effectiveness through the creation of more flexible, modular, and powerful IT infrastructures.

Obstacles to the adoption of open systems include users' lack of awareness and current investments in proprietary systems, the immaturity of several open systems technologies, and the confusion caused by competing standards efforts. Nevertheless, the open systems "train" has left the station and it will not turn back. Users within DoD need a "standardized" standards planning process for IT. Lack of such a process has resulted in planning and implementation delay. All functions face the challenge of migrating to standards-based technology while prudently managing the installed base of proprietary systems through the interim period towards a standards-based target architecture.

The new IT architecture

IT architecture plays a key role in making IT user requirements work. Traditional computing environments based on proprietary products and isolated data processing systems have resulted in a costly, poorly integrated, and hard-to-change infrastructure in most organizations. IT architecture should provide a coherent blueprint by which systems are integrated into an interoperable whole.

A new, volatile, strategic and operational environment demands new capabilities from IT that traditional computing environments cannot deliver. Rather than upgrading their current environments, leading organizations are setting out on a course of migrating to a new environment based on the new technology paradigm. Research shows that functions that are retooling invariably conclude that a new network architecture can only be achieved through the adoption of standard interfaces and components.

The result is the emergence of the *"standards-based"* architecture. Such a function-owned architecture can include the vendor-independent standards associated with open systems. A standards-based architecture will include a migration strategy from interim proprietary standards to open standards.

The standards-based architecture is based on a number of components that do not appear in traditional technology plans. These include architecture principles, definitions of generic components, and a set of industry standards supported by products and technologies that adhere to those standards. It defines reusable and interchangeable architecture components that promote flexibility and modularity in the architecture.

What is architecture?

An analogy can be useful in understanding what an architecture is and why it is important.

IT architecture is the underlying framework that defines and describes the IT platform required by a function to attain its objectives and achieve a functional vision. It is the structure given to information, applications, and organizational and technological means—the groupings of components, their interrelationships, the principles and guidelines governing their design, and their evolution over time.

Like planning for a building

An IT architecture is analogous to the architecture for a building. The plans for a building include provisions for the various services to be offered in the building, such as electrical power, plumbing, communications wiring, stairwells, and elevators. They must also provide the overall design of the building (i.e., its construction specifications, how many floors there will be, the look of the exterior and interior walls, etc.).

An architecture plan must also consider zoning laws, regulations and standards for building usage, such as set back from the street, orientation on the lot, and blending with the existing environment. It must also consider the ingress and egress, general work patterns of the desired tenants, layout of the equipment that may be housed in the building, and the type of construction material needed to meet the usage requirements of each area of the building.

The architecture must ensure that components of the building fit together to meet the needs of the prospective tenants and the surrounding environment. It must also have the ability to evolve with the changes that time may bring, perhaps the need for expansion or for alternative uses.

The architecture does not, however, concern itself with details such as the specific color of carpet a given tenant may want, or exactly how each person's desk will be oriented, or even how each individual office space may ultimately be built out to suit the tenants' cosmetic or work flow needs.

Rather, the architecture concerns itself with providing a flexible, adaptable infrastructure to meet these varying needs without tearing down the building and starting over. This is accomplished by adhering to solid principles of architecture design, by developing a set of blueprints (or frameworks) for the building's appearance and layout, and by setting some basic standards for the construction teams to follow as they implement the plans.

Typically, the architecture does not specify particular vendors or suppliers for the components of the building. Instead, it provides flexibility by setting standards for the components, which may be met by one or more suppliers. In this way, competition among alternative suppliers allows the architect and construction teams to keep costs in control while minimizing the risk associated with sole source relationships.

Of course, as the construction begins, some specific decisions will have to be made about vendors as well as the details of construction for a given tenant. In the construction planning phase, the architecture still forms the framework for decision making, but more detailed plans will have to be developed for each tenant's specific requirements. Here, the cost of materials, durability requirements, specific equipment locations, and office layout must be considered. A detailed design must be developed with specific cost estimates, time to complete, and vendors to be used. This goes beyond architecture planning but must remain true to the architecture principles and blueprints for the overall building.

The analogy

There is a direct analogy in the IT area for each of the points discussed above. The architecture principles for the building define the overall style of the building and its general characteristics, given its envisioned usage. Similarly, the IT architecture principles are the foundation for decision making about the general style of computing and technology usage for the company.

For example

"The building will be a skyscraper, no more than 60 floors, envisioned for general office usage, of steel and glass construction with non-opening windows, in the style of a monolith, with integrated underground parking, pre-wired for high-speed telecommunications on every floor, with external elevators facing the bay."

With these principles, one gets a fairly good idea of the kind of building this will be, and some of the constraints that will be placed on vendors who may qualify to work on the project as subcontractors.

In IT, the principles provide a similar mechanism for defining the kind of information systems we will have.

"To the extent possible, similar business functions will be supported by common systems, which will support all physical locations. These systems will be run locally, within each plant location but will be maintained and updated from a central location.

The systems will be developed within an industry standard environment and will be interconnected for data sharing via a series of interconnected telecommunications networks, which will communicate using industry standard protocols. Access to all systems will be via intelligent workstations connected to the network and using a set of common user interface standards."

A starting point for detailed design and system construction

Just as the artist's rendering and a general description of a new building's characteristics are not enough for the construction crews to do their work, the principles of an IT architecture are not sufficient to allow the system designers and implementors to construct appropriate information systems.

In the case of the building, realistic scale models of the structure are developed to aid the architect in envisioning how the various subassemblies of the building will all fit together. Blueprints of the mechanical, electrical, structural, and other aspects of the building will also be developed.

These blueprints and associated specifications define the overall infrastructure of the building, envisioning the needs of the classes of tenants who are likely to occupy the space. The basic services of the building are defined and placed within the infrastructure, usually according to a set of well-defined industry standards and codes.

There is a direct correlation in the development of IT architectures. The principles are used to guide the development of models and associated specifications for the way the organization will use IT.

IT architecture models are like an architect's blueprints

The four views of an IT architecture (the way work activities are organized, the information needed to perform the work, the automated systems that capture and manipulate the information, and the technology environment within which these automated systems run) are analogous to the detailed architecture blueprints and specifications for the subassemblies of a building as described above.

As with the building blueprints, the IT architecture models must anticipate the classes of users, their location within the organization, the type of work they must do, and the anticipated need for automated systems in these locations. It must do so without knowing in advance all the details of each automated system that may be needed by these users in the future.

The bottom line on architectures, for buildings and for IT, is providing a minimum, but rigorous, set of guidelines and standards that will allow the building (or information systems) to be developed in a way that will allow the most flexibility for the tenants (or system users) while constraining the detailed designs enough to ensure that the desired style and characteristics of the building (or the computing environment) are maintained over time.

With these principles, the style of computing and communication is defined in enough depth to allow appropriate detailed design work to begin and vendors to be selected.

What is IT architecture planning?

So, with the prior analogy as a backdrop, we define architecture planning as the art and science of transforming a functional need for computer-based systems into a planned and organized framework that supports integration and enables systems design and delivery.

Architecture planning proceeds on three fronts:

- The definition of a commonly accepted framework around which architecture decisions can be based
- A clear definition of organizational responsibilities and planning procedures is required to ensure architectural integrity
- Each major systems project requires a level of architecture planning based on these guidelines and organization to address specific system requirements.

A new approach to architecture planning

The need and opportunity to create a functional IT architecture based on standards are both new. Similarly, the new functional imperatives and the new technology paradigm demand a new approach to technology planning and migration.

Traditional architecture planning only focused on application and data design to support individual applications. Methods were based on techniques that limited scope and created hard boundaries. Solutions were evaluated and chosen based on specific vendors and products. Criteria emphasized functional fit and cost, not architecture considerations.

The new SBA planning approach is quite a different proposition. The new approach to SBA planning deals with both the structure and style of computer-based systems. It requires the definition of architecture components or "building blocks" and ways to describe the relationship among architectures. IT architecture provides that often elusive link between identifying a strategic opportunity to apply computer solutions and choosing the best available solution. Most importantly, it describes the standards upon which these building blocks are assembled.

Multiple views of the architecture

The IT architect must serve a number of communities of interest. It is therefore necessary that the architecture framework support the communication needs and viewpoints of these various interest groups.

Standards-based architecture is also multifaceted. While constantly relating to strategic functional requirements, architecture must reflect four different views of the transformational change involved in using IT. These four views are:

- **Work organization view.** How will the planned system impact work activities (nature and magnitude), change skill requirements, affect functional operating locations, and eliminate or reduce manual support systems?
- **Information view.** What information bases are required to operate the function? What forms and volumes of information are involved? What relationships between the information bases must be provided? What access and security controls are required?
- **Application function view.** What types of application functions are required to support the transformed organization and associated users? How will functions be grouped and interfaced? What usage levels are anticipated?
- **Technology view.** What types of technology services are required and how should they be distributed to various types of technology platforms? How will these services and platforms be networked, and what standards and guidelines are required to support integration?

The four views of the integrated architecture are shown in Figure 1-1.

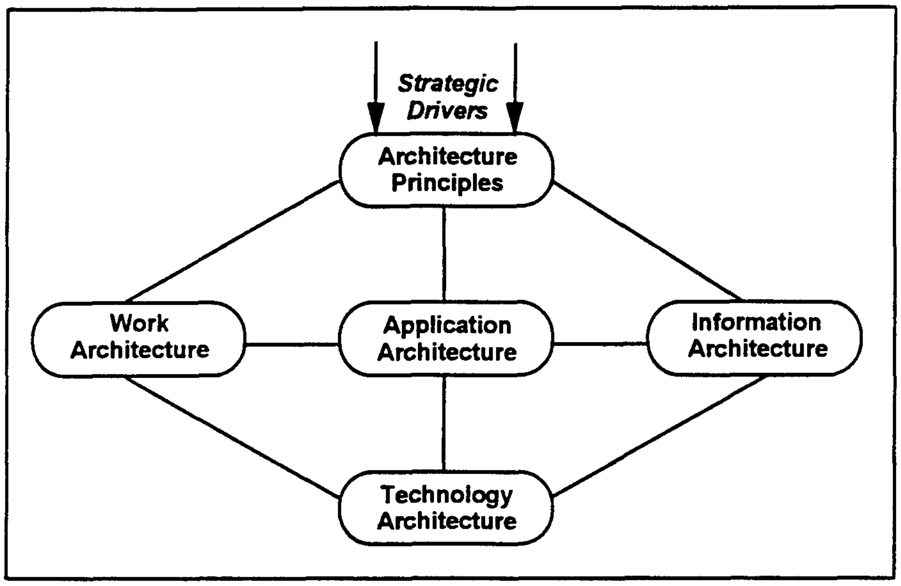


Figure 1-1. Architecture Modeling Framework

The architecture principles, and their upward link to the strategic drivers of the enterprise, provide the basis for reflecting the strategic use of IT—the domain of the executive group and strategic functional planners. They are used to show how the operation of the function will benefit from the transformation changes enabled by IT. They provide the functional strategists' views of the architecture and are used to drive out the predominant architecture principles.

Work organization view

The work organization view describes the major operations that are performed by work groups in support of functions. It defines the types of work (logical working units) in terms of the types of workers (classes of IT users) and types of work locations (places where the functions of the organization are carried out).

The work organization view should be independent of line organization design. Many traditional IT solutions were tailored to specific line organizations, resulting in hard boundaries and inflexibility. Work organization modeling recognizes the realities of "networks" of individuals and their supporting automated and manual systems. It supports the team concept, the multiple roles (or team memberships) that individuals can have, and recognizes that teams can be composed of members who work remotely from each other.

It also should recognize external users and external functional locations. Key external constituencies (e.g., legislative organizations such as Congress) and suppliers are obvious candidates. Employees working from home office locations or while traveling should also be considered for inclusion.

The work organization view helps to describe the before and after impacts of technology on the organization. It becomes the basis for detailed redesign of work processes, communication programs, and user training to address change management requirements.

Information view

The information view describes the information used by the organization and the relationships among collections of information (subject databases).

It is important to include all forms of information and types of media in this view. Again, placement and distribution to working locations in support of user and application access is a key consideration.

Application view

The application view shows which functions of the organization can be supported by IT applications. It provides a high-level description of these application opportunities. It also shows logical dependencies and relationships among application opportunity areas.

This view defines the scope and interfaces of applications and provides the basis for detailed design. It identifies specific work groups and users of applications, their relationships to information, and their placement or possible distribution across types of locations and technology platforms.

The application and information views are used in tandem to define the targeted applications and information that will support the organization. Together they drive the requirements for technology.

Technology view

Technology views are used to describe the enabling infrastructure. To provide the necessary linkage to the work organization, information, and applications architecture views, the technology view can further be described in terms of some generic building blocks. These include: Generic Application Environments (GAEs), Generic Technology Environments (GTEs), and Generic Technology Platforms (GTPs). These are described in Appendix D.

Architecture modeling frameworks and their uses

The architecture modeling framework defined above has been developed to support the IT architecture planning process and related deliverables. The modeling framework has many uses:

- It is used to explain the meaning and concepts of architecture planning, particularly the multiple views and purposes that a complete IT architecture must serve.
- It provides a basis for describing the current IT architecture and assessing its strengths and weaknesses.
- It is used to describe the target IT architecture. It provides all the necessary components to describe the required architecture that best supports the strategic directions of the function. It provides the generic components from which specific target environments and their interrelationships can be modeled. In particular, it can be used to determine common requirements that exist within and across organizational units. These common requirements provide the basis for defining infrastructure. The resulting infrastructure views then provide the basis for defining standards and guidelines for component design and acquisition.
- Finally, the modeling framework is used to guide the major steps in a migration strategy to bridge the current and target architectures. Consequently, it can be used to update the progress toward the target as well as to adjust architecture plans to reflect changes in functional direction or unforeseen technology advances.

In most organizations, IT architecture planning is a relatively new endeavor. Early attempts usually focused on only one or two of these four views, with little regard for the others. It is important that standards-based architectures reflect a balance of these four views of their relationship.

As a result of the newness of architecture planning and the accompanying high rate of change, the "science" component of architecture is incomplete and inconsistent. Businesses typically lack the common language and disciplined approach necessary for architecture planning to serve its practitioners and communities of interest.

Goals of an architecture

Given this, an architecture must address three goals:

- Provide a means of cost effectively organizing information and its technologies to support the organization's objectives
- Improve the effectiveness of IT in delivering new capabilities to the organization
- Facilitate continual evolution of the IT infrastructure and solutions over time.

The approach outlined herein attempts to do just that—provide a step-by-step process that may be used in a typical function. It may be amended, adopted, and modified to conform to the standard IT planning approaches that may already exist in the enterprise.

The questions it addresses are:

- By what process can we define a standards-based architecture that meets our functional vision?
- How do we get from here to there?

Large enterprises, for example, cannot discard large investments in proprietary mainframe and mid-range applications and hardware. They cannot suddenly switch to an operating system such as UNIX merely because it is more "open." Likewise, users who have a considerable investment in PC-DOS machines cannot adopt X/Windows overnight if the changeover requires conversion of 10,000-20,000 workstations already field deployed.

A multivendor environment is one characterized by hardware and software diversity. These distinct and unique environments are generally required to work together at the function level. This requires a high degree of technical and operational coordination. In most organizations, this occurs on a "patchwork quilt" basis at best.

The standards-based enterprise focuses on standards-based architecture in a "diverse" technology environment because it enables these diverse environments to interoperate effectively. A key characteristic of an open systems environment is the critical need for "rules of the road" or regulated standards. For open systems to work effectively in an organization, the standards-based organization must have a method for developing a enterprise-wide standards-based architecture.

Traditional IT planning approaches

To understand the new approach to architecture planning let's begin by assessing the inadequacies of existing IT planning methodologies.

Many organizations have tried using a traditional IT planning model. Frequently these IT planning approaches, while interesting exercises, are never implemented in the traditional organization. The reasons for this lack of implementation are organizational, functional, or technology changes that occur before action is taken. These "strategic" plans have typically been built on 3- to 5-year time horizons, with linear project plans that take several years to complete. The fundamental problem is that the planning processes do not reflect the reality of today's operational or functional environment.

Traditional planning approaches, when conducted properly, model a function or organizational entity and outline programs for applications, data, and technology platforms. The output from these planning exercises is a document that often represents the culmination of many person years of planning across a function. In many organizations, such plans are frequently relegated to the filing cabinet and soon become fossilized "shelf documents." The plan's creators are frequently the only personnel that have actually read the detailed plan. Generally, traditional plans include an executive summary that receives wide circulation but, because the larger plan is not read, many unanswered questions are left about what to do next when it comes time for implementation.

Such plans are typically difficult to modify as the function, the organization, or the technology changes. Getting original plan participants to participate on a meaningful but mammoth update effort is difficult. Traditional technology platform programs outlined in the plan become obsolete

12-24 months later as IT vendors introduce new technology or, as is often the case, delay introduction of technology forecasted for adoption in the traditional plan document. The following diagram illustrates this IT planning dilemma:

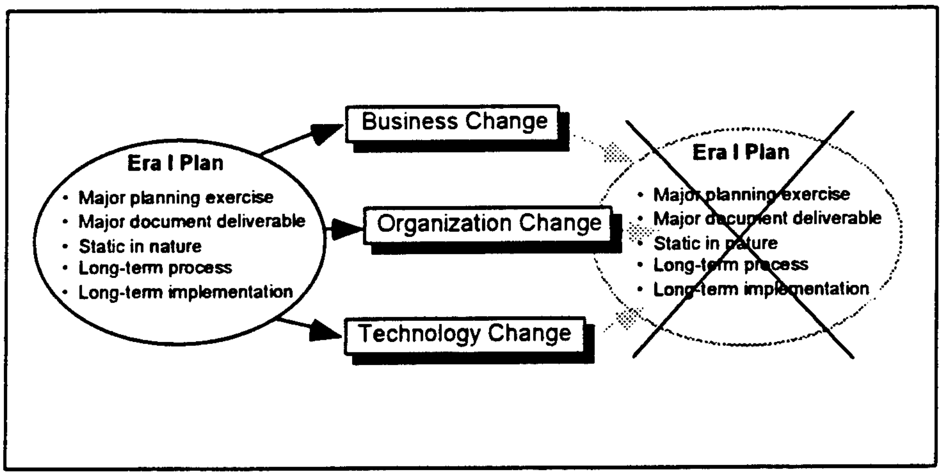

Era I Plan
• Major planning exercise
• Major document deliverable
• Static in nature
• Long-term process
• Long-term implementation
Business Change
Organization Change
Technology Change
Era I Plan
• Major planning exercise
• Major document deliverable
• Static in nature
• Long-term process
• Long-term implementation

Figure 1-2. Traditional IT Planning Dilemma

Perhaps the weakest link in traditional planning models is implementation. Because of the various functional, environmental, and organizational issues described above, many traditional IT plan efforts are never put in place. These traditional planning approaches typically break down in the manner in which they approach defining technology standards. This activity is simply regarded as an added and unnecessary step in developing architecture. It does not allow for a decoupling of the technology from the "architecture" in the context of standards. By comparison, standards-based infrastructure modeling assumes that the organization and technology will change; indeed, change is the only constant.

Standards-based planning vision

Standards-based organizations place a premium on a flexible, standards-based architecture. They acknowledge today's reality that all business functions are competing in time and that the static, linear planning model that traditional planning methodologies represent is obsolete. Standards-based organizations recognize that relationships between functions, organization, and technology are often not aligned but seemingly discontinuous.

Who "owns" the vision?

With the dispersion of control over IT into the functional units out of the "glass house," the IT planning agenda itself is increasingly driven by the end-user side of the enterprise rather than the traditional IT organization. The "ownership" of the traditional IT plan has changed because the "stakeholders" have changed.

Standards-based organizational stakeholders are operational users, component units, and suppliers. This is a major shift from the traditional IT planning context when IT professionals owned and sponsored the IT agenda. Increasingly, end users are asking their IT professionals to provide value for the investment of the last decade.

In the past, major application projects have been delayed by several months or years, which has resulted in a major negative impact on operations. For better or for worse, end users are demanding results *now*, with no excuses or "technical mumbo-jumbo" for nonperformance.

Operational or functional users are increasingly setting the direction for IT planning. The decentralization of functional units and the parallel and attendant introduction of end-user technologies, such as LANs, personal computers, workstations, and network technology, has only accelerated this trend. The logic is simple: "The IT folks can't deliver, so we functional unit professionals will have to make it happen."

The need for a shared process

Despite the fact that functional users are increasingly taking control of the IT agenda, successful standards-based architectures can only be built when the planning process itself is driven by functional and IT professionals *working together* to integrate the dynamic "counter pulls" of diverse functional initiatives, organizational work flows, applications vehicles, networks, and technology platforms together in an overall strategy with a focused thrust. Any standards-based planning process and effort must take this critical fact into account. Little will be accomplished if standards implementation occurs independently and for its own sake. The key measure of the merits of standards implementation is the degree to which standards cumulatively provide significant functional value to the function.

The following diagram illustrates some of the various tensions at play with IT planning today:

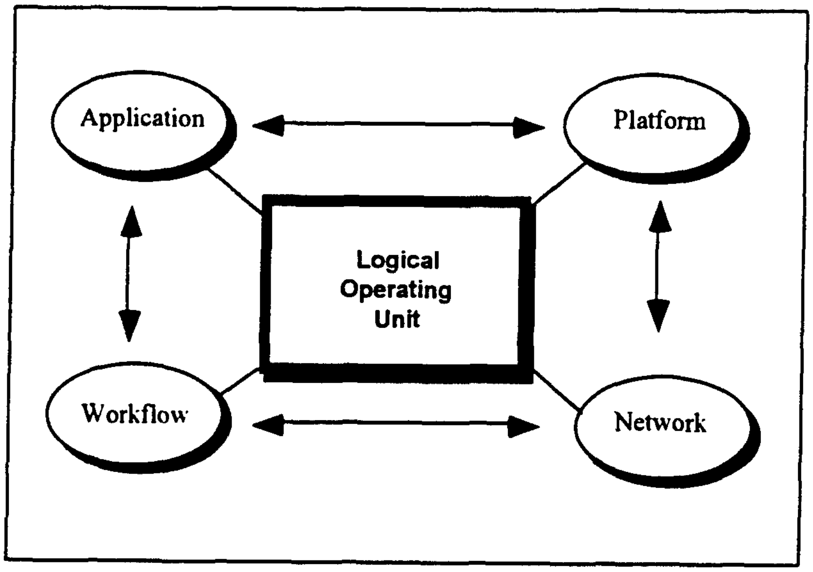


Figure 1-3. IT Planning Tensions

Traditional vs. standards-based planning characteristics

Several key characteristics distinguish standards-based organizations from traditional IT organizations in their functional and IT planning activities:

Traditional IT Planning	Standards-Based Planning
Long-term vision, long-term payoff	Long-term vision, short-term payoff
Major function-wide "data gathering" effort	Function fast-path "process"
Primarily defined and "owned" by the IT organization	Primarily "owned" by the functional unit
Proprietary vendor architecture owned by vendors	Standards-based, open architecture owned by the user
Vendor leverage over user is high	User leverage over vendor is high
Functional unit input limited	Functional unit focus central
Based on coherent "linear" functional strategy	Based on discontinuous, chaotic functional realities of today's "fast cycle" global marketplace
Static document-oriented deliverable	Project-oriented deliverable payoffs
Obsolete when organization or technology changes	Continuously modified on quarterly basis
Typically defines functional drivers, applications and data and specific proprietary hardware/ software solutions	Defines architecture and standards with room for entrepreneurial improvisation in implementation

Figure 1-4. Traditional Versus SBA Planning Characteristics

The remainder of this SBA Guide explains the steps one should take to develop a standards-based architecture.

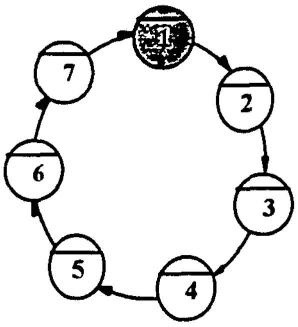
This page intentionally left blank.

Section Two: Initiation and Architecture Framework

Contents **Page**

Figures

Section description



This section describes the overall process that is followed to initiate the SBA planning activity and to develop the first major deliverable—the *Architecture Framework Document.* The following are the key aspects of this phase:

- Project initiation and positioning within the enterprise
- Development of a general definition of the open systems development and architecture environment
- Definition of an architecture vision for the future
- Consideration of a general review of architecture design alternatives
- Identification and documentation of issues underpinning the architecture vision.

Project initiation is a critical key to ultimate project success and, as such, is discussed first.

Project initiation

Project initiation provides for a smooth transition from initial project planning to the architecture framework phase of the project. It is essential that the project initiation step be explicitly defined and executed for, without it, the project will not have the firm foundation needed to withstand the inevitable rough times. Architecture projects, particularly at the enterprise level, uncover all of the basic insecurities of the host enterprise. Sensitivities are revealed, sacred cows are questioned, and political issues are raised. If these foundation issues are not dealt with and clearly agreed upon by all involved parties, the project will falter when these periodic storms hit. The facilitator needs to be aware of all these issues and realize that open lines of communication from the very beginning of the relationship are absolutely essential to the success of the project.

By their nature, all architecture engagements are different. As a result, an explicit project initiation step is a key to success. The phases, tasks, roles, and responsibilities will be affected by the culture of the enterprise, architecture work that may have already been done by the enterprise, the commitment of resources the enterprise is willing to make (or conversely insists on making), the preconceived notions the enterprise has about what an architecture project entails, and a host of other factors too numerous to list. Project initiation allows all involved parties to agree on the customization of the basic SBA planning approach taking all of these factors into consideration. It then allows specific decisions about resourcing and time frames for the agreed-upon tasks. A clear-cut project plan emerges and the first stage of the plan is kicked off.

The project initiation step is not completed until a plan has been laid out in enough detail for the enterprise to know exactly what is expected at all points along the way. Obviously, not every single workshop, interview, or background session will be scheduled to the day and minute, but the necessary events of the early stages of the engagement should be locked in during project initiation. Also, the critical project infrastructure issues (CSFs) must all be resolved.

Almost all of the work of project initiation revolves around the key issues of establishing a mutually agreeable resourcing strategy and allocating those resources to tasks that will result in deliverables and time frames with which all parties can live. Then, of course, the key early tasks in the plan will be kicked off.

Architecture Work Group

The core team that will be involved in the SBA project from beginning to end is the Architecture Work Group (AWG). This is the group of four to six mid-tier managers and IT personnel from the functional areas. This team will be responsible for facilitating the SBA process, for developing the overall project plan, for securing appropriate participation by key knowledge workers, and for ensuring that all documents specified in the project plan are completed.

Architecture Steering Committee

The key to success in this phase depends on the ability of the AWG to help the participants develop a shared understanding of the problems and opportunities related to the existing environment and then to establish a coherent framework for solving these problems over time—*building a shared vision and direction.* While it is the objective of every planning exercise to develop this vision, it is frequently not achieved for a very simple reason—key players were not involved in the process.

Because of this, it is critical that an Architecture Steering Committee (ASC) be formed. This group should be composed of a mix of functional area and IT professionals. Its size and makeup will differ depending on the scope of the SBA effort. If there is a question of team membership balance, it is preferable to err on the side of too many functional area professionals. It is paramount that all stakeholders be involved in the team—this includes any individuals or enterprises with key influence or other "political" power within the functional area.

The *Architecture Framework Document* is developed by the AWG. Together with key knowledge workers (these are the subject matter experts with specialized skills or knowledge that work on an as-needed basis with the AWG), this team becomes the core entity for developing the rest of the SBA project.

The bulk of the research for the *Architecture Framework Document* is conducted by facilitating "fast-path" workshops and interviews with key functional and IT personnel. The team produces evolutionary drafts of the document until all of the stakeholders enthusiastically endorse it.

A multistep process is an effective way not only to identify the central issues underpinning a standards-based architecture but to help develop the architecture principles that will guide the rest of the effort.

It is important to note that this phase of the standards-based implementation cycle is of a direction-setting nature. During this effort, a general understanding of the current environment is developed and a high-level definition of the current architecture direction is rendered. Time should not be spent uncovering minute technical details. That work is better left for subsequent steps of the process.

Objectives

It is important to produce a comprehensive *Architecture Framework Document* that is easy to understand and that engages executive commitment. It is also important that the document be function oriented—addressing issues that are key to the success of the functional area(s) included in the effort.

The AWG should avoid focusing solely on technology and the application development environment. Executive staffs will often dismiss a technical document because they see little benefit in defining technology for technology's sake; however, a document explaining what technology can do to help the enterprise achieve its mission is sure to get executive attention.

Scope

The scope includes all aspects of the enterprise that may have an impact on the future use and deployment of IT—the work of the enterprise and the way IT may be used to support it. Key business drivers are defined as well as the issues surrounding current technology. Workshop and enterprise change-related activities are the primary vehicles by which the *Architecture Framework Document* is produced.

Personnel in each functional area within the enterprise are interviewed by the AWG. The purpose of these interviews is to:

- Discuss the basic mission of the functional areas
- Identify areas for improvement in current practices
- Begin to determine possible ways that information technology can be used to better support the enterprise.

The AWG then synthesizes the findings of the interviews. The results of this synthesis are a set of architecture principles. These principles are then put to the test. They are voted on and discussed with the ASC. This meeting provides a vehicle for key stakeholders to discuss and agree on how the enterprise should proceed with this very important SBA task.

The principles presented in this deliverable will serve as guidelines for developing the plans that will ultimately become the IT architecture for the enterprise.

Deliverables

An *Architecture Framework Document* that contains:

- Enterprise mission/vision
- Strategic drivers
- IT principles
- Key issues that will impact development of the target architecture.

The major deliverable of this phase is the *Architecture Framework Document*. It is recommended that this document be brief in nature, "Executive Summary" in design, and as highly visual as possible. A sample outline for this document is included in Appendix I.

The central objective of this document is to provide a broad understanding of the IT architecture vision. If the document is produced successfully, all key stakeholders will possess an "ownership" of the effort.

Critical success factors

- Identifying shared interests
- Establishing the ASC and chairperson (effectively the "system owner" team)
- Establishing the AWG and primary contact (effectively the "system manager" team)
- Establishing the larger community of knowledge workers who will participate, either in interviews or workshops
- Establishing the mechanism to officially kick off the engagement for all of the participants identified above and for the enterprise as a whole
- Providing initial orientation to the architecture development process for the ASC, the AWG, and the community of knowledge workers who will directly participate
- Supporting the executive level of each functional area within DoD
- Establishing a shared vision
- Providing a communication vehicle for promoting the vision of the architecture design
- Assuring key knowledge worker commitment and participation
- Agreeing on how, when, and to whom project status will be reported
- Procuring and setting up workspace and tools for the facilitator(s) and the AWG.

Constraints

Many enterprises have never formally developed architec-ture principles. The absence of these principles is a definite constraint to the work team, which relies heavily on such documents in defining the mission and vision of the enterprise.

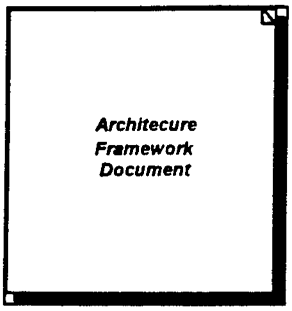
- Commitment and participation of executive staff (ASC)
- Availability of existing source material.

Management must be solicited to dedicate knowledgeable personnel to the effort (at least until the necessary vision statements and principles are created) or the project is doomed to drag on indefinitely, while the AWG attempts to define this starting point.

Task list

- Initiate project and AWG team building
- Form ASC
- Define interview process
- Conduct interviews
- Analyze existing information
- Evaluate existing data-gathering processes
- Optimize those processes to ensure timeliness and accuracy
- Reconcile interview data with existing information
- Draft and circulate principles for principles workshop
- Conduct principles workshop
- Review final principles with ASC
- Create *Architecture Framework Document* outline
- Assign writing, reviewing, and editing tasks
- Draft *Architecture Framework Document*
- Circulate *Architecture Framework Document* for comments and review
- Review *Architecture Framework Document* with ASC
- Finalize and publish *Architecture Framework Document.*

Creating and publishing the deliverable



This phase will vary widely in terms of the calendar time required for completion based on culture, individual schedules, etc. Ideally, when conducted on an intensive basis, this phase can be completed in approximately 4 weeks. However, most enterprises require about 2 months to complete the outline, draft, and final document. The document simply goes through several iterations before approval by the ASC. The process is as follows.

With the ASC as a quality check, the AWG can begin to conduct the interviews necessary to gain insight into the business drivers within the function. If done properly, these interviews can also serve the purpose of promoting the architecture project throughout the enterprise.

Senior executives and key "thought leaders" within the enterprise should be interviewed. Because of the high exposure that this activity represents, it is important that the interviewers be well prepared prior to scheduling the first round of interviews.

It is suggested that a set of essential questions be developed jointly across the body of interviewers. This helps the interviewer anticipate underlying issues and problems before actually interviewing key personnel—thus minimizing the potential for failure. Figure 2-1 highlights general questions to be asked. These questions can be more detailed depending on the scope of the SBA endeavor.

Existing models and principles

To expedite building the architecture framework, the team should review any existing business, work organization, application, and information models, as well as current architecture principles for background. There is no need to "reinvent the wheel" if such materials exist. The models provide input and background to the AWG.

Reconciliation and principles workshops

The result of interviews and secondary research of existing material is the development of a set of draft principles. As the effort progresses, principles workshops are held. Each workshop addresses specific topics such as applications, standards issues, database strategies, and communications.

Sample List of Questions

1. What are your responsibilities today?
2. What are your current and long-term priorities? What stands in your way?
3. What are the most critical elements for success in your job?
4. How can technology be used to help you succeed?
5. What has been your experience in technology projects in the past? What has made them successful? Why have they failed?
6. What improvements can be made to make your work environment more productive? Can technology be used?
7. Would you be willing to commit resources to improving the use of technology in your area?
8. Who would you recommend we talk to next regarding the use of technology in your area? Would toy help us schedule a meeting?

Figure 2-1. Interview Questions for Input to Architecture Framework

The purpose of the workshops is to reconcile the views and principles with the information uncovered in the interviews. A group of architecture principles is developed. It is typical for a group to develop 30 to 40 different principles for an enterprise's architecture. A sample principle taken from the USMC project is shown below. In addition, a more complete description of how to develop architecture principles is included in the SBA Guide as Appendix A.

Architecture principles are **statements of preferred architectural direction or practice.** They are simple, direct statements of how an organization wants to use information technology in the long term for five to ten years. They establish a context for architecture design decisions across an organization and help translate business criteria into a language that technology managers can understand. Each principle is accompanied by a statement of the rationale behind stating the principle and a statement of the principle's implications.

Figure 2-2. Definition of an Architecture Principle

Principle

Where feasible, the USMC will use Commercial-Off-The-Shelf (COTS) and Government-Off-The-Shelf (GOTS) application components and systems rather than develop them internally.

Rationale

The use of COTS and GOTS applications and components should lead to an environment of increasingly interchangeable parts. This kind of environment should be more cost effective and efficient than custom development, because multiple "customers" are sharing in the development and maintenance costs. For similar reasons, training and implementation costs should be reduced. The time frame from concept to implementation should be reduced by taking advantage of tested and operationally proven applications and/or application components. Finally, the risks normally associated with custom development (e.g., scope changes, budget overruns, missed target time frames, etc.) are significantly reduced.

Implications

- A process for evaluating and selecting COTS and GOTS applications will be needed. This process must accomplish at least the following tasks:
  - Identify user requirements which can be satisfied by purchasing standard components.
  - Consider if changing the current functions and processes would enable the purchase of standard system components without adverse effect on operational performance.
  - Analyze whether the USMC's customization needs can be accomplished outside the purchased standard component rather than inside it. In so doing, the Marine Corps could subscribe to the vendor's ongoing maintenance releases.
- Some BPR may be needed to align the business process with available COTS or GOTS applications.
- A set of standards and measurements for matching a standard component's functionality with the user requirements should be developed. For example, the standard might state that only systems or components which satisfy 80% of required functionality should be considered for purchase.
- A repository of available COTS and GOTS applications will be needed. This repository will need to accommodate the definitions of the applications and/or application components as well as any predefined interrelationships among the applications.
- Finally, using COTS and GOTS systems and components will make the USMC reliant on those vendors for maintenance and upgrades. Therefore, a vendor qualification process must be undertaken to assess the potential longevity in the marketplace of vendors of prospective packages.

Figure 2-3. Sample USMC Principle

Effectiveness measures

- Degree of consensus achieved with principles
- Acceptance of draft *Architecture Framework Document*
- Amount of rework required
- Management participation
- Awareness of the effort.

The overall objective of this phase is to provide a summary document that is easily understood by business managers and IT personnel alike. It is therefore important that the deliverable be a functionally oriented (rather than technically oriented) document and focus on key issues of importance to the functional area(s).

The work team will be measured against its ability to develop a document that the enterprise "buys into." Granted, this is a very subjective measure. However, it is the only one that really matters at this stage in the SBA project—buy-in is the name of the game.

For this reason, minimal rework alone does not guarantee quality work. Sometimes minimal rework points to a lack of management commitment to the effort.

Therefore, effectiveness can only be measured by the combination of variables listed above. The team will know if the results of its effort are falling on deaf ears, if few people within the function know about the SBA project and even fewer senior managers pay it due.

Technology and tools required

- Dedicated war room for team meetings
- Word processing and graphic presentation packages
- Microcomputer and telecommunications capabilities
- Principles templates (see Appendix A)
- *Architecture Framework Document* outlines (see Appendix I).

To truly expedite the effort, a project "war room" should be established. It should be equipped with a white board and markers for brainstorming, PCs for preparing the document, a table and a set of comfortable chairs for conducting meetings and interviews, and plenty of work space so that the team can get the job done.

The AWG should be equipped with word processing, spreadsheet, and graphics presentation packages so that they can develop the *Architecture Framework Document* easily. If possible, the team should be connected to each other via a network so that the work files can be passed from writer to reviewer more efficiently.

In some of the more sophisticated environments, the work room is staffed with a secretary who can take messages, help with the typing, and assist with the document preparation work; however, this is not a prerequisite.

Staffing skills required

- Group facilitation skills
- Interview skills
- General functional area knowledge and IT technology background
- Project management skills
- Writing and presentation skills.

The key to this effort is the solicitation of management support for the effort. Therefore, it is essential that a good group facilitator is used—one who can manage group dynamics, understands the SBA process, and can keep the work team on track.

This kind of individual is present in most enterprises; however, many firms feel more comfortable getting their facilitation expertise from outside the concern—outsiders tend to be more objective and are less likely to sway the team for personal gain. Figure 2-4 highlights some essential facilitator skills.

Although the facilitator is important to this effort, he/she does not a work team make. The work team must be staffed with people who possess the qualities listed above, or the effort could be in jeopardy. For this reason, work team candidates should be screened prior to project inception—just to make sure the right people are available for the job.

List of Essential Facilitator Skills

- Knowledgeable project manager
- In-depth understanding of SBA process
- In-depth understanding of automated tools used in SBA process
- Expertise in team building
- Expertise in managing group dynamics
- Ability to communicate in both business and technical terms

Figure 2-4. Essential Facilitator Skills

Completion criteria

- Interview schedule completed
- Draft principles document
- *Architecture Framework Document* deliverable
- Management acceptance.

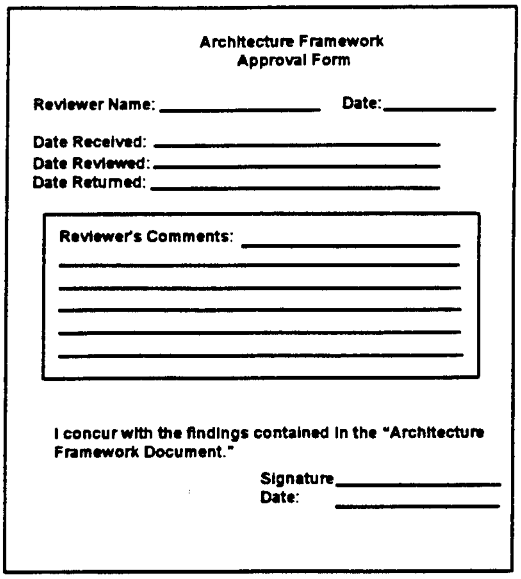
Ultimately, this phase is completed when the ASC accepts and signs off on the *Architecture Framework Document.* While the other items listed above are important milestones, the work is not considered complete until all committee members "own" the deliverable.

For this reason, it is important for the team to establish a sign-off procedure that ensures full committee approval. Many times enterprises will establish a sign-off procedure that assumes acceptance with no formal reply. This should be avoided. Figure 2-5 illustrates a typical Architecture Framework Approval Form for committee sign-off.

A process that requires a written signature has proven to be very effective. ASC members will pay more attention to the effort because they want to understand and be in agreement with what they are signing.

Issues

- Training required
- Executive participation
- Current workload of work team members
- Consulting support required
- Subject matter expert availability.



Architecture Framework
Approval Form

Reviewer Name: ______________ Date: __________

Date Received: ______________________________
Date Reviewed: ______________________________
Date Returned: ______________________________

Reviewer's Comments: ______________
____
____
____
____
____

I concur with the findings contained in the "Architecture Framework Document."

Signature______________
Date: ______________

Figure 2-5. Architecture Framework Approval Form

As mentioned throughout, executive commitment and the availability of key personnel (or key knowledge workers) is essential to the success of this effort. However, there are other issues that an enterprise must face to ensure a quality deliverable from this phase.

The need for training and consulting support is often overlooked by enterprises excited about establishing a standards-based architecture. While every function is different (in the skills and talents that its personnel possess), most require the initiation of training in the planning technique presented here.

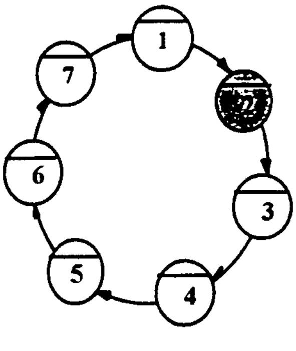
For this reason, most enterprises use consultants to provide the necessary training and to drive the SBA effort—at least until the enterprise becomes self-sufficient (usually after one or two successful SBA pilot projects have been conducted at a functional area level).

Section Three: Baseline Characterization

Contents **Page**

Figures

Section description



This section describes the overall process that is followed to conduct a high-level characterization of existing work organization, information, applications, technology, and standards. This activity includes:

- Reviewing general cost, performance, and security issues related to the baseline architecture
- Developing a framework for characterizing the current environment to help the WAG organize its thinking
- Documenting the characterization of the current environment in a *Baseline Characterization Document.*

Objectives

To create a report that characterizes the existing architecture of the enterprise.

Many organizations have undertaken enormous baseline efforts sometimes requiring many months, if not years, to complete. The detail that would take years to develop is not necessary–characterizing the existing situation in just a few months of elapsed time is the goal.

Without the insight that a baseline characterization provides, it is difficult to develop truly effective implementation plans needed to lead the organization into its chosen target architecture. A clear view of the existing IT architecture allows identification of opportunities for change and a migration plan for implementing those opportunities. Without this view of the existing situation, there is the risk of devising a target environment that is very difficult or impossible to implement.

The SBA process is designed to be "fast path" in nature. That means that traditional long-term inventory efforts will not be appropriate if the task is to proceed quickly and deliver results. While large and timely data collection efforts yield more accurate data, time is sacrificed for accuracy. If a branch of service or entity already possesses much of the baseline data, then most of the work effort should be spent on characterizing the current environment with a high-level description. The difference between a good and bad baseline effort is the degree to which the baseline *characterizes* the current environment accurately. The recommended approach is a generic baseline versus a detailed specific baseline.

Scope

The enterprise that is being modeled (e.g., a branch of service, a subset of a service, the entire DoD):

- Existing views of physical and logical environments can be used if readily available.
- Task teams can be formed to develop information about the current environment, if no formal data exists.
- Matrices for categorizing work, information, application, and technology platforms as well as cost frameworks can be used.
- Descriptive security classification should be applied to each application and the technology environment reviewed.

The AWG should set their sights on conducting a baseline effort that characterizes the current environment rather than conducting the most accurate inventory effort. This is not the same activity as a massive inventory effort! In practice, and as a rule of thumb, 80 percent of the information used in an architecture design activity derives from 20 percent of the data collected. It is therefore inefficient to spend time collecting the last 20 percent of the data when 80 percent is sufficiently accurate in characterizing the current environment. Figure 3-1 illustrates the data collection payoff dilemma all AWGs face.

Fundamentally, all IT architectures are built upon existing technology platforms. In the end, an IT architecture represents how the given sets of existing technology platforms are used and structured and the attendant functionality they deliver for the individual, the work group, the function, or the enterprise.

The task of evaluating and designing a new or alternative architecture requires that the AWG have a convenient method by which it can characterize the current architecture. After the AWG has created a baseline of the existing architecture, its relative merits and shortcomings can be examined. With a baseline in place, assuming the function seeks to improve upon the existing architecture, the team will be able to develop a target architecture and an all-important migration plan to assure its successful implementation.

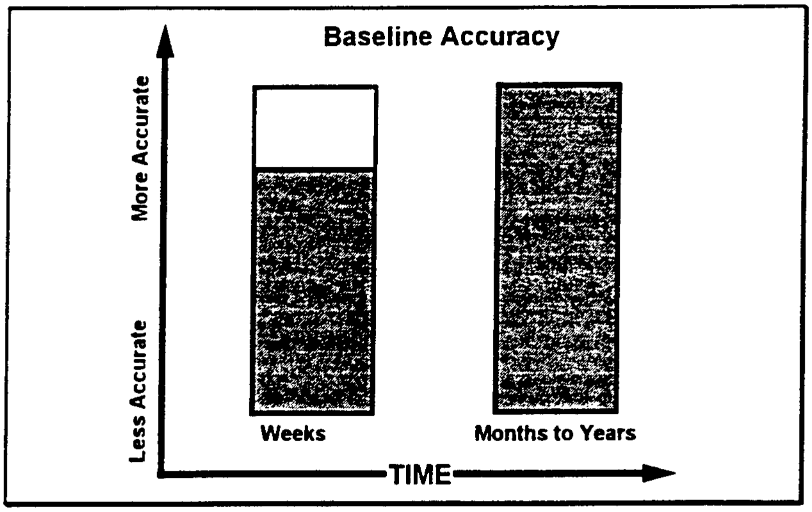


Figure 3-1. The Data Collection Payoff

Baseline elements

A number of elements should be reviewed as inputs to the overall *Baseline Characterization Document.* These include:

- Work organization view
- Information view
- Application view
- Technology view
- U.S. Department of Defense *Technical Reference Model and Standards Profile* (TAFIM, Volume 2) is a framework with which to characterize current profiles in place in different parts of the overall model
- Security design document, which specifies the security plan for the organization. It contains information about such issues as security policy, accountability, security assurance, and security documentation as outlined in the U.S. Department of Defense Trusted Computer System Evaluation Criteria [DoD 5200.28 STD, December 1985].

Deliverables

The major deliverable of this phase is the *Baseline Characterization Document.* The DoD recommends that this document be brief in nature, "Executive Summary" in design, and as highly visual as possible. The idea is that this document will be used by a large number of individuals and organizations as will all deliverables produced during the architecture development activity.

Appendices should contain the results of the baseline data gathering, while the body of the document should contain key conclusions and analyses. This document should show readers "the forest" rather than focus on "counting trees." A sample outline for this document is included in Appendix I.

Critical success factors

- Commitment of resources to develop inventory information
- Trained leadership with experience in fast path baseline efforts
- Communication vehicle for reporting inventory information
- Key knowledge worker availability.

A key critical success factor is that the senior management of the function understands, endorses, and enthusiastically champions the SBA project. In a time of shared DoD resources, this means committing DoD personnel to work on the project for dedicated periods of time. Therefore, a premium must be placed on time and doing the baseline effort quickly.

As stated in the previous section, the ASC, composed of representatives from both the business and IT departments, will act as the "project owner." This committee is the conduit between the AWG and the rest of the function. It is key that the ASC makes all concerned organizations aware of the vital nature of the baseline effort and secures cooperation from the same when required.

Constraints

Availability of existing architecture input in readily accessible form.

Many organizations have never formally developed or created baseline models. The absence of these models is a definite constraint to the AWG, which relies heavily on such documents in defining the current environment.

However, these background materials can be developed quickly when the right people are engaged in the effort. There are people within the organization who understand what information exists and the level of effort required to collect data appropriate to the task at hand. Management must be solicited to dedicate such knowledgeable personnel to the effort, at least until the necessary architecture views and principles are created, or the project is doomed to drag on indefinitely.

Task list

The baseline characterization process follows the basic steps listed below. A key step in the process is primary data gathering in the form of workshops with key knowledge workers in various operational areas. Workshops are conducted with one or more representatives from the host organization.

- Initiate baseline task team—identify AWG and task groups
- Define inventory scope, effort, and milestones
- Develop application, technology environments, security, cost and platform classifications, and data collection instruments (templates and tools)
- Assign inventory data-gathering tasks
- Review findings and synthesize results
- Produce first cut *Baseline Characterization Document*
- Conduct management review of *Baseline Characterization Document*
- Refine *Baseline Characterization Document*
- Distribute *Baseline Characterization Document* to ASC for comments and review.

The AWG conducts the overall baseline activity and is responsible for producing the *Baseline Characterization Document.*

Data collection

The AWG should appoint a small subtask group to conduct a baseline effort that characterizes the current computing environment. This task group conducts a technical inventory of the organization's existing technology infrastructure. Inputs to this process will vary widely from organization to organization based upon the quantity and quality of documentation available. Business, process, and data model documents may also be used as input. Physical diagrams, logical diagrams, tabular inventory, and financial budget data will also be valuable.

One recommended source for baseline data is the *Defense Automation Resources Information Center (*DARIC).

DARIC maintains an extensive set of database repositories that inventory installed hardware, software, and data related to management of information technology within the DoD. At the same time that the DARIC resource may be used to provide useful baseline information to AWGs, DARIC may also be used to review technology components that might be valuable for reuse. It is highly recommended that the AWG meet with DARIC personnel to obtain a detailed understanding of DARIC's capabilities and resources.

It may become necessary for the baseline task group to assemble and conduct workshops to derive data from the organization when it is not otherwise readily available from DARIC or other conventional sources.

Overview of the baseline activity

To establish a baseline architecture, an inventory of the existing computer and communications hardware, system software, and application systems must be compiled.

The inventory is not intended to be exhaustive. Do not spend an excessive amount of time and effort on collecting the information. Eighty percent accuracy is sufficient to establish the basic structure of the baseline. The primary goal in collecting this baseline inventory is to establish the overall existing architecture structure and a high-level view of its robustness on a number of levels, including user satisfaction, strategic significance, and technical quality.

Baseline inventory

The baseline inventory will be compiled by completing a series of worksheets or templates. A complete set of templates, used in the baseline assessment, is included in Appendix B. The templates cover all of these categories:

- Existing work functions and processes
- Technology platform inventory
- Applications inventory
- Initial application assessment
- Various affinity (cross-reference) matrices showing the interrelationships of the various components of the baseline architecture.

Work functions and processes

This inventory should include all business functions and the key processes included within the function. For each function, the mission should also be identified. These functions and processes should be cross-referenced to other components of the baseline architecture in the following ways:

- Functions to data groupings
- Functions to applications
- Functions to locations.

Technology platforms

This inventory should include all components of the computer processing and communications environment, including the following information:

- Type of platform (in terms of the generic technology platforms defined in Section 3 of the SBA Guide) and outlined below:
    - Workstation
    - Output/input peripheral
    - Local area network (LAN)
    - LAN server
    - Wide area network (WAN)
    - Network interface device
    - Concentrator/multiplexer/switching device
    - Storage devices
    - Mid-range processor
    - Large processor.
- Vendor name and model (e.g., IBM 3090, IBM 486 PC, Sun Sparcstation). Also include the capacity characteristics in terms of throughput and associated storage (memory and access to separate storage devices).
- Specific technology environments (standards) supported in the following areas:

  - User interface
  - Operating system
  - Communications management
  - Database (and/or file) management
  - Transaction monitor
  - Document management
  - Distribution management (e.g., E-mail, electronic data interchange)
  - Conferencing management
  - Development services (compilers, languages, and tool support)
  - Repository services (for systems management and construction, including data dictionary support).

- Platform owner (i.e., who has the budgetary ownership or responsibility for this platform).
- Platform manager (i.e., who has the day-to-day operations responsibility for the platform).
- Platform location (i.e., the physical locations of the platform, address, building number, and/or other designator which will uniquely define the location).

Initial application assessment

As a part of the collection of the existing inventory, an initial assessment of the application systems should be gathered from key application users. System developer/maintainers should also give their assessment of the more widely used applications.

An initial assessment of each application is needed according to the following criteria: user satisfaction, strategic value, and technical quality. As part of the analysis process, after all templates have been returned, these criteria will be mapped in the following pairs on four-quadrant matrices to allow a high-level determination of the recommended disposition of each application:

- User satisfaction versus strategic value

- Technical quality versus strategic value
- Technical quality versus technical evolution.

Mapping attributes to platforms

One of the key activities of this phase is the development of a description of the current environment. This activity must be simple to accomplish. Most organizations have technology platforms in place that handle existing applications. These platforms, more often than not, are supported by proprietary technology.

A range of technology platform categories are provided that will be used in the baseline effort. The criterion for platform definition is that it must be offered in the marketplace as a product. It must be viable, proven technology that is available in the marketplace and one that users can purchase and implement in the present. These technology platforms include:

WS

- **Workstations.** Any device ranging from a fixed function or dumb terminal to a high-end workstation capable of complex calculations and graphic requirements (e.g., 3270 terminal, PC, SUN workstation).

O/I Per

- **Output/input peripherals.** Any device that outputs or inputs electronic data (e.g., laser or line printer, image scanner).

LAN

- **Local area networks (LANs).** Operating system protocols associated with local area network solutions (e.g., Ethernet, Token Ring, Starlan).

LAN Server

- **LAN servers.** Network operating system software and hardware attached to LAN networking solutions that allows routing, file storage, and user application services (e.g., LAN Manager, Novell, Banyan, 3Com, Netframe Super-Server).

WAN

- **Wide area networks (WANs).** All network services offered by public network providers such as public and virtual private switched voice, switched and dedicated data, gateway and enhanced service offerings (e.g., AT&T, MCI, U.S. Sprint, Telenet, Internet, IBM Information Network, Tymnet, Telenet, etc.).

Iterfce.

- **Interface devices.** Any device that provides a major bridge or switch between environments (e.g., TCP/IP router, DEC router, LAN bridge).

Con./Mux
Switching

- **Concentrator/multiplexer/switching devices.** Any device that performs a concentration function, a multiplexing function, or a switching function (e.g., IBM 3705, a NET T-1 multiplexer, an AT&T PBX).

Storage

- **Storage devices.** Any traditional magnetic or optical storage device (e.g., floppy disk, magnetic tape, optical disk).

Mid-Ran.
Proc.

- **Mid-range processors.** Historically known as the "mini-computer," this increasingly blurring category includes any processor manufactured for mid-range processing (e.g., IBM AS400, DEC VAX, HP Spectrum).

Large Proc.

- **Large processors.** Traditional mainframe category historically dominated by IBM, UNISYS, and Amdahl. Supercomputers, such as Crays, are included at the high end of this category.

Technology platform attributes

The various generic platform classifications described allow a baseline inventory to be made of the existing architecture. As IT technology changes, so will these categories.

Each platform listed above may be thought of as having various attributes. By categorizing existing platforms and defining their constituent parts, a standards-based current architecture may be defined and examined in a baseline exercise. It may then be used in subsequent steps to define the target architecture. Figure 3-2 illustrates these various platform attributes.

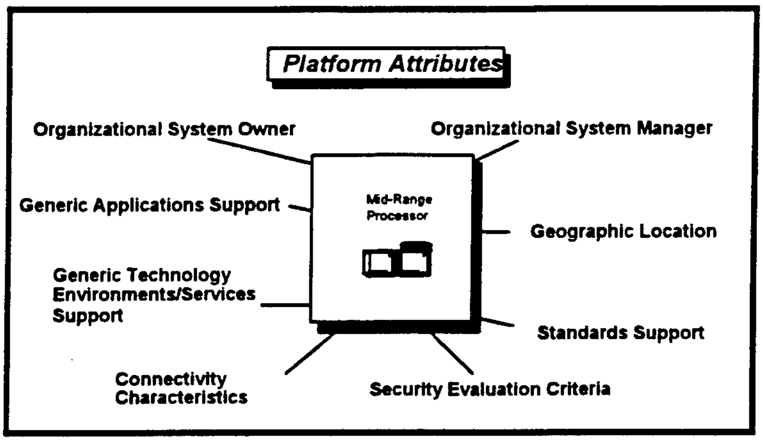

Platform Attributes
Organizational System Owner
Organizational System Manager
Generic Applications Support
Mid-Range Processor
Geographic Location
Generic Technology Environments/Services Support
Standards Support
Connectivity Characteristics
Security Evaluation Criteria

Figure 3-2. Platform Attributes

Each of these platforms:

- Has a specific system owner(s) with a DARIC reference number
- Has a specific organizational system manager
- Supports an application or application suite and thus serves a role as a generic application support environment or "GAE"
- Provides a technology role for an overall architecture through the provision of services as a generic technology environment or "GTE"
- May be classified in terms of its security evaluation criteria as outlined in *Trusted Computer System Evaluation Criteria Summary Chart* (p. 109) of the U.S. *Department of Defense Trusted Computer System Evaluation Criteria* [DoD 5200.28 STD, December 1985]
- Supports various standards, be they proprietary or open in nature, and are built on either de jure or de facto standards
- Has connectivity and interface characteristics with other technology platforms
- Has specific cost performance characteristics associated with its technology life cycle
- Has a specific physical environment.

The following diagram illustrates how the technology platform attribute model may be used as a model for a baseline platform—in this case, a mid-range processor.

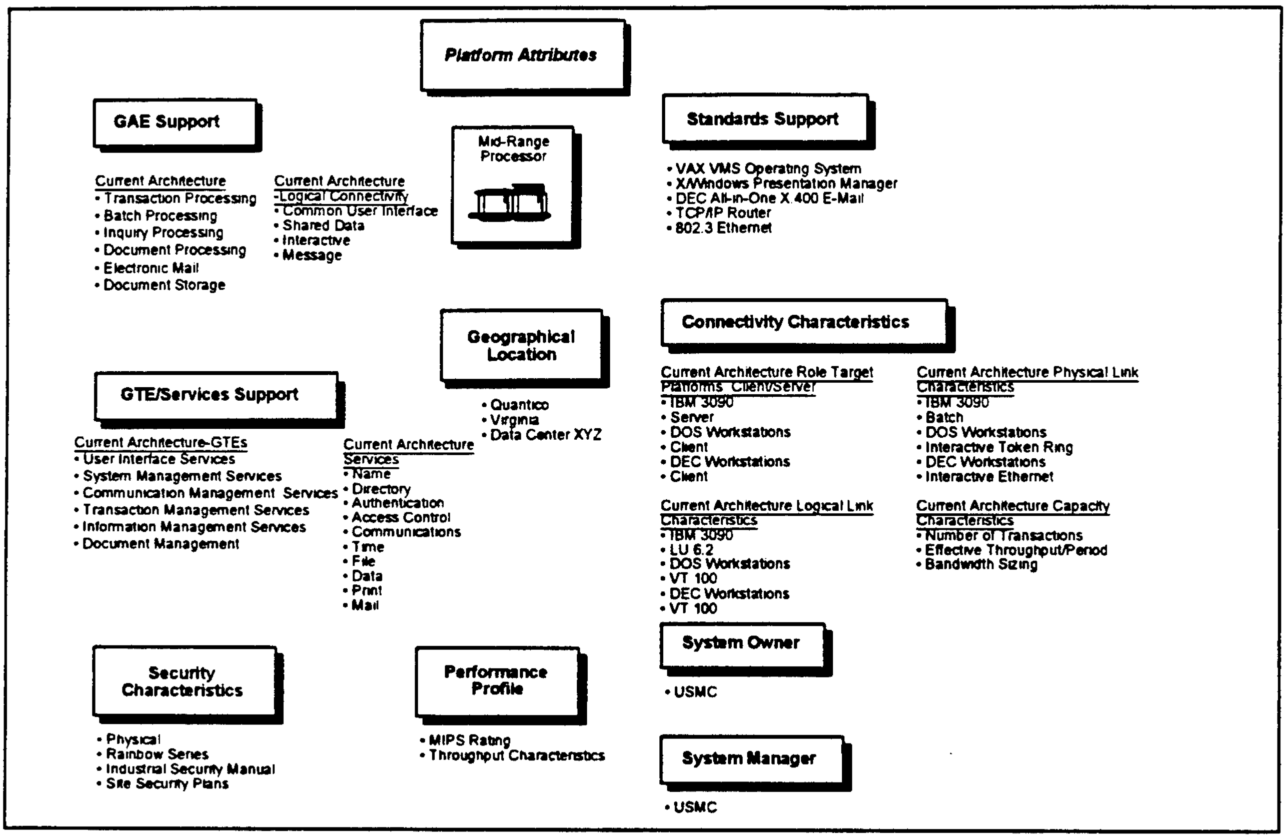
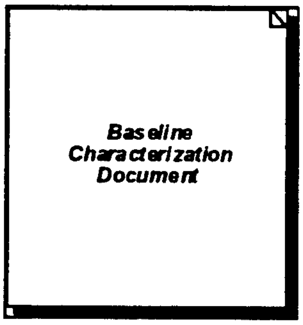


Figure 3-3. Platform Attributes Examples

Creating and publishing the deliverable



The key deliverable out of this phase is the *Baseline Characterization Document.* The sole objective of this document is to characterize the current environment and to highlight systematically the profile and attributes of the current architecture. The baseline will be used as input to the migration options phase where it will be compared to the target architecture. This comparison will be used to identify necessary projects to achieve the vision of the enterprise.

The *Baseline Characterization Document* provides a total picture of the current state of architecture. This phase will vary widely in terms of calendar time required for completion based on enterprise culture, individual schedules, etc. Ideally, when conducted on an intensive basis, this phase may be completed in 8 to 10 weeks.

However, most organizations require about 3 months to complete the outline, draft, and final document. The document should go through several draft iterations before being approved by the ASC.

The overall objective of this phase is to provide a summary document that is easily understood by business managers and IT personnel alike. It is, therefore, important that the deliverable be a business-oriented document and focus on key issues of importance to the function.

Effectiveness measures

- Management acceptance of task deliverable
- Comprehensive global characterization of existing environment
- Amount of existing inventory data that is reused
- Speed of task execution
- Extent that document is accurate as measured by degree of acceptance (and percentage degree of completeness).

Technology and tools required

- Word processing and graphic presentation packages
- Architecture team room for meeting
- Spreadsheet tools and/or user friendly, personal computer-based database packages for inventory logging
- Baseline templates (see Appendix B).

The AWG should be equipped with word processing, spreadsheet, database, and graphics presentation packages so that they can develop the *Baseline Characterization Document* easily. A key aspect of this activity is the development of data collection templates to streamline the project data-gathering exercise. Once these have been created, the rest of the baseline effort is more mechanical than "creative."

Staffing skills required

- AWG with baseline experience and high familiarity with existing environment to be baselined; for example:
  - An inventory specialist who provides input to Arms database
  - Network administrators
  - System managers

  - Data administrators.
- Interview skills
- Writing and presentation skills
- Organizational data collection knowledge
- Familiarity with word processing, presentation, spreadsheet, and database packages that run on most popular personal computers.

The key to this effort is the solicitation of management support for the effort. Therefore, it is essential that an AWG leader is selected to facilitate the baseline effort—one who can manage group dynamics, roll up his or her sleeves with the team and participate, and who understands the SBA process and can keep the work team on track.

Completion criteria

- Inventory scope and deliverable defined
- Inventory completion deadline met on time
- Management acceptance of deliverable
- Completion of *Baseline Characterization Document.*

Ultimately, this phase is completed when the ASC accepts and signs off on the *Baseline Characterization Document.* It is important that all the ASC members as well as the AWG agree that this document is a characterization of the current environment.

The team should obtain a sign-off that ensures full ASC approval. This was described in the Architecture Framework section.

Issues

- Workload of work teams
- Availability of existing inventory data
- Successful amount of data collection in short time frame
- AWG understanding of level of effort and fast path approach
- Core team to remain the same.

The need for resources on this task is crucial to project success. The overall AWG may be at its highest level of headcount during the baseline effort.

Given the severe resource limits that are currently the norm in the DoD, we recommend that the AWG draft members on a temporary duty basis for the baseline effort. The "baseline draftees" may then be demobilized and released or be assigned to the target architecture phase upon completion of the *Baseline Characterization Document.* However, the core AWG members remain the same throughout the overall project period.

The ideal profile for an "enlisted" AWG member drafted to conduct baseline work is an individual who possesses a sense of urgency and the ability to work on a "fast path" basis to ensure project success.

Keep in mind that the baseline effort is not intended to determine an action plan for solving the ills that it uncovers (such plans will be developed during the implementation planning phase of the project). Instead, the intent is to simply define the current environment, which will act as a logical launch point for subsequent phases of the SBA process. What's next, however, is to define the target environment that the organization seeks to embrace over the next few years.

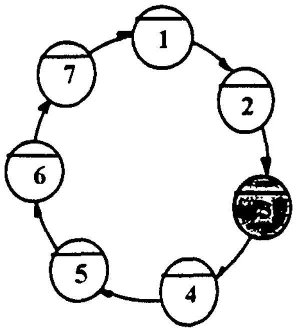
Section Four: Target Architecture

Contents **Page**

Figures

Figures (cont'd)

Section description



This section describes the overall process by which the architecture framework is extended by the AWG. These issues for you to approve includes:

- An extension of the vision defined in the *Architecture Framework Document*
- A description of a desired future architecture
- An identification of what can be extended from the current environment into the target environment.

Objectives

To develop a *Target Architecture Document* that specifies the profile and attributes of the new technology environment and highlights the key opportunities for improvement over the baseline. The new architecture need not be developed based on cost-effective and "business-case-based" criteria. The real world constraints of cost/benefit analysis and cost justification will be introduced in the migration options phase of the SBA process.

At this step in the process, it is desirable to define a target architecture that can be used to achieve the vision of the organization in all of the architecture views and, especially, the work architecture. Ultimately, constraints will come to bear on the funding of each project that is needed to achieve the target but, for now, it is sufficient to flesh out the target to identify the full spectrum of what is needed to achieve the vision of the organization.

Inevitably, the architecture that is implemented will be a blend of the baseline and the target, with architecture principles as the foundation stone. Sometimes, an organization cannot migrate to the target without either disrupting the quality of service provided to the user base or expending an inordinate amount of resources to get there. Therefore, it is important that the team take the time to outline a set of alternative architectures that may become an interim target until the ultimate target can be legitimately reached.

Figure 4-1 depicts an overall framework within which the AWG can operate to develop the target architecture deliverable. Each view of the target architecture has some overlap with aspects of the other views (see Figures 4-2, 4-7, 4-9, and 4-11 below). This overlap supports the argument that we are developing a single, integrated architecture. As we proceed through the remaining discussion of the target architecture development process, we will frequently refer to this meta-model in order to remain focused on the key aspects of the task at hand.

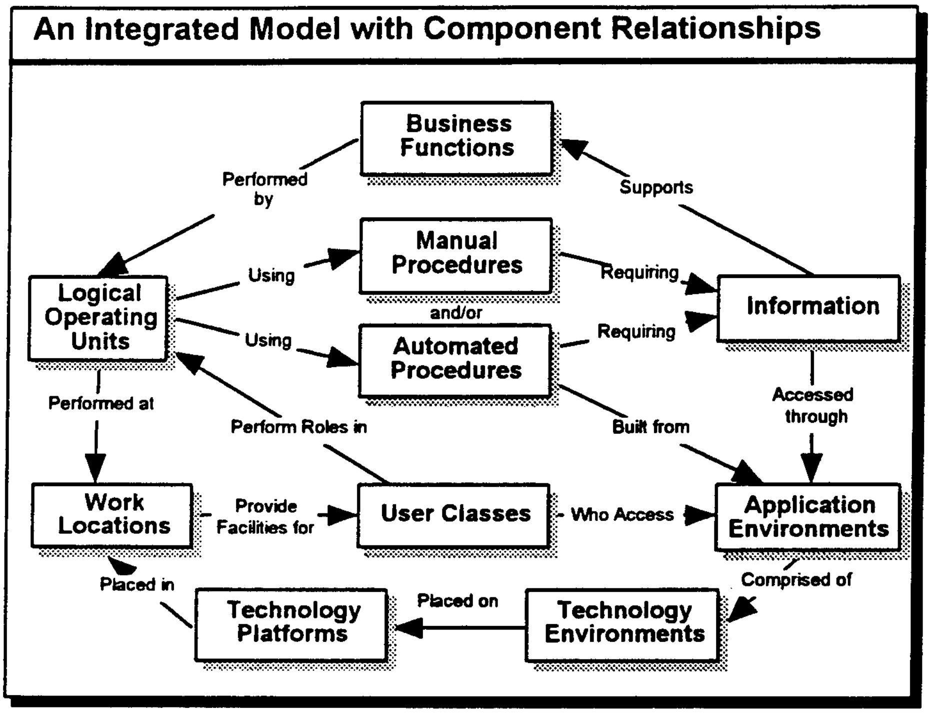


Figure 4-1. Integrated Model of Four Architecture Views

Scope

The entire enterprise, as defined, including:

- Work organization
- Information
- Applications
- Technology.

Many planning methodologies have a process within them that advocates the creation of a target architecture. Frequently, however, the target architecture is too general and is of little value (e.g., "We will use a relational database management system for client files"). At the other extreme, the target definition can be too product specific to be considered truly open (e.g., "We will use IBM's DB2 for our client files").

The key to creating a quality blueprint document is defining the target architecture in such a way that it would remain open and flexible over time as technology, products, and infrastructure evolve.

Deliverables

A *Target Architecture Document* that describes:

- Target architecture with the four views defined, as well as the key interrelationships across the views. A sample outline for this document is included in Appendix I.

Critical success factors

An AWG that has:

- A combined general understanding of the current functions and processes of the enterprise
- Experience in long-term functional area and IT planning
- A practical understanding of the tradeoffs between functional issues and technology
- A working knowledge of systems development and maintenance
- An effective communications vehicle between the ASC and the AWG.

It is extremely important to staff the AWG with seasoned professionals. To do otherwise can be disastrous. Team members must come to the planning table with experience in business and IT planning. They must also have the political sensibilities to understand the limitations inherent in their work environment.

Constraints

- Lack of functional area and technology vision in the AWG
- Lack of full-time commitment to the project by management for key knowledge worker participation in workshops
- The team's inability to comprehend the potential of the SBA process.

Task list

- Initiate task
- Define target architecture environment planning process
- Assign team to review the *Architecture Framework Document*
- Develop the work view of the architecture
- Develop the information view of the architecture
- Develop the applications view of the architecture
- Develop the technology view of the architecture
- Create the draft *Target Architecture Document*
- Conduct review with ASC
- Finalize *Target Architecture Document*
- Distribute *Target Architecture Document.*

Reviewing the principles

In the first phase of developing the SBA framework, the key component of standards were developed—the architecture principles. All target architecture work is based upon these principles. Principles are similar in nature to a federal constitution. They become the central document against which all deliberate and explicit standards-oriented policies and guidelines are developed. In this phase, the target architecture principles are extended into more specific models of the four views of an integrated target architecture.

Detail the target with four views of the architecture

The target architecture defines the IT environment needed to support the organization over the agreed-upon planning interval (usually 5 or more years). Its aim is to achieve the vision for the future outlined in the *Architecture Framework Document* for all four views.

Work architecture

This work view of architecture is developed by identifying specific classes of users within the business environment (e.g., executives, planners, administrators, engineers, recruiters); business locations (e.g., headquarters, sales office, plant, warehouse); and a logical representation of the business functions that are required to deliver products and services. This "logical" unit of work is called a logical operating unit (LOU). These three basic components of the work view will ultimately be mapped to the applications (i.e., automated procedures), manual procedures, and information required to support the work. This linkage helps to integrate the work view with the other views of the target architecture.

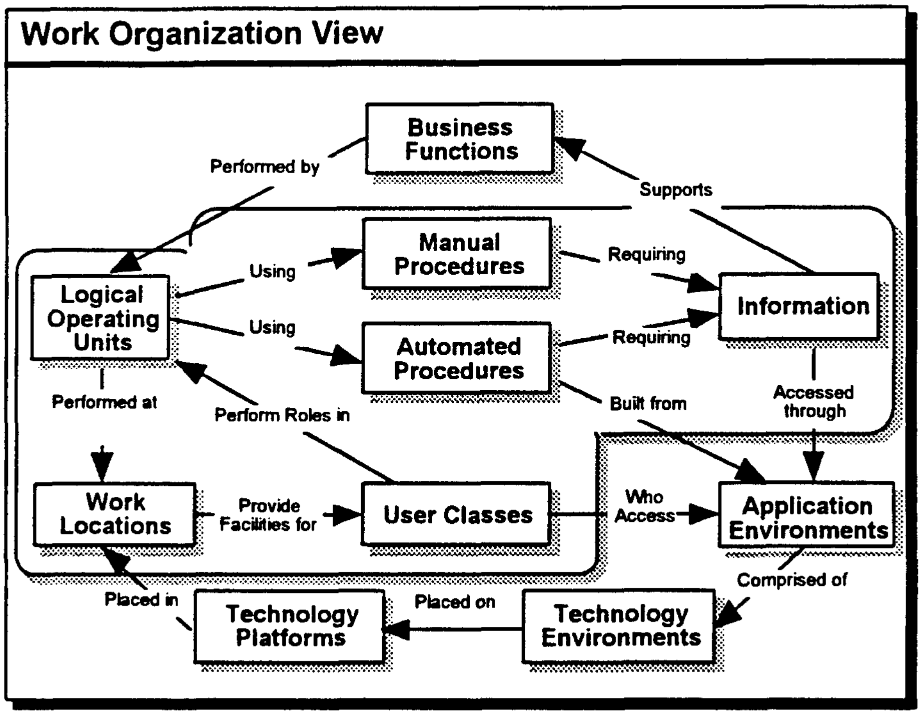


Figure 4-2. A Work View of the Architecture

This "logical view" of work will be independent of today's line organization and/or physical locations. It will be the "pure" view of the work required to deliver products and services. This pure view can then be mapped to the existing physical organization and locations, allowing opportunities for IT automation, integration (of systems and functions), and/or work redesign to be identified.

Other views of architecture will impact the work view

The other three views of architecture (information, applications, and technology) may have an effect on the work view. As the definition of the future view of work proceeds, the process should include discussions of the information required by each LOU, the kinds of systems (applications) that may be needed, and the kind of technology that might support such systems. Obviously, at this early stage of architecture development, these views of the target architecture do not yet exist, but we can do some early, high-level analysis as a way to help us validate the LOUs in the work view. We want to capture the essence of these discussions to feed into the process of developing the detail of these other views of architecture.

The process facilitator's responsibility is to ensure that the team does not get bogged down in detail during these discussions and, more importantly, to ensure that a broad enough view of the future is taken.

Although there can be multiple ways to legitimately segment an enterprise's business, discussions generally yield 10 or fewer "Major Business Areas." The names for these major areas should not be confused with similar names for existing organizational units since they represent generic business functions, not existing departments or work groups. Start the process of defining these major business areas with a brainstorming session with executives and key knowledge workers from the enterprise. The facilitator should go into these sessions with the following generic major business areas "in their back pocket." These generic areas are used to guide the discussion if it begins to stray or if the teams get stuck and need a little help:

- Planning
- Selling
- Buying (raw materials acquisition)
- Manufacturing (or whatever the "core business" is)
- Delivery (product distribution)
- Collecting
- Support (including such things as finance, human resources, administration).

Each major business area is then broken down into its logical components of work, or LOUs. As with the major business areas, LOUs are not associated with the current organizational structure, its labels, the person performing the work, or any physical location.

Every LOU (see Figure 4-3) must provide a service and may have suppliers of products or services. It must be possible to measure its contribution; if not, it is probably not a LOU but an activity within a LOU. Each LOU is defined by the output (or service) for which it is conceptually responsible and the activities it must perform to achieve this result. A LOU always delivers its product or service to "customers" within the enterprise or within external actors beyond the boundary of the enterprise being modeled. A customer within the enterprise is always another LOU. A customer beyond the boundary of the enterprise is an external actor (e.g., "true" customers, suppliers, other Government agencies, parent organizations). Usually, a LOU will also be supplied with information or materials.

As the work organization view (i.e., a network of LOUs) is being developed, it is important to define the way the work should be partitioned and defined, not necessarily the way it is today. This network of LOUs should reflect the most effective and efficient way for the work to be done in the future. To achieve this, the LOUs themselves and their interrelationships will have to be developed, tested by applying various scenarios to them to see if they hold up, and refined as necessary to optimize the organization of the work within the enterprise. We may think of the major business processes within an enterprise consisting of the execution of one or more LOUs in sequence. In this sense, the LOUs are the major steps along the way in a business process.

A key point to remember is that a LOU may participate in more than one business process at varying points in time. Regardless of how many business processes a LOU participates in, its purpose, and the work activities that are executed to achieve that purpose, remain constant. In this way, the enterprise can develop policies, procedures, and supporting systems and tools for the most stable aspect of the business, the LOUs and, by definition, these policies, procedures, and supporting systems and tools will effectively support all business processes, which are made up of various combinations of LOUs.

The next step in developing the work view of architecture is to map the LOUs to classes of users who will perform the activities of the LOU. These user classes themselves are also logical in nature. As such, a physical employee of the enterprise may belong to one or more user classes.

Characteristics of Logical Operating Units

- The Logical Operating Unit (LOU) is the fundamental building block for defining architecture models.
- The LOU is defined primarily by its role in the production or delivery of one or more products or services within the operation of the enterprise.
- LOUs have distinct roles and responsibilities (no overlaps, redundancy, ambiguity, or gaps).
- It can be related to the overall contribution; the requirement for the LOU is clearly understood.
- Its performance can be measured.
- A LOU must have a customer and provide a service (it also may have a supplier).
- LOUs are independent of:
  - The organizational structure and departmental names
  - The degree of automation
  - Who does the work
  - Where the work is done.

Figure 4-3. Characteristics of Logical Operating Units

One or more user classes can be mapped to a given LOU, signifying that these user classes will perform at least one of the work activities of the LOU. A user class will not be related to the LOU if it only receives or passes information from or to the LOU. The user class must actually be the one performing one or more of the work activities defined within the LOU.

The final piece of the work view of the IT architecture is the concept of logical work locations. All of the "types" of work locations will be defined, regardless of how many physical locations may be involved. For example, "Base" might be a logical work location, while there may be multiple physical locations that contain this logical work location, such as Honolulu, Albany, and New Orleans. Figure 4-4 describes the process of identifying logical work locations.

Logical (and Physical) Work Locations

- Just as we wish to insulate our systems from the effects of organizational changes, we wish to insulate systems as much as possible from the effects of changing physical locations.
- To do this, we identify a set of Logical Work Locations. Similar to the way user classes allow us to categorize employees in terms of the roles they play, in a generic sense, the Logical Work Location concepts allow physical locations to be characterized in terms of the roles they play.
- There can be many Physical Work Locations that contain a given Logical Work Location.
- A given Physical Work Location may contain more than one Logical Work Location.
- In all cases, the Logical Work Locations should be set up to allow a reasonable mapping of Logical Operating Units (LOUs) against these locations.

  This mapping gives the architecture model the necessary linkage back to the user class. It also allows for a forward mapping to Physical Work Locations. These linkages are key tools in determining where application systems and supporting IT platforms will be located within the enterprise.

Figure 4-4. Logical (and Physical) Work Locations

With the logical characterization of work operations, users, and locations, supporting systems can be built that are completely independent of today's physical constraints. This provides the ability to develop the most flexible and adaptable systems.

As the user classes and logical work locations are mapped to the LOUs, additional refinements may be made on the LOUs themselves. Discussing who performs the work and where the work is performed will frequently lead to better ways to partition the work. No part of the work view of architecture is "cast in concrete" until all of the dimensions (LOUs, user classes, logical work locations) and their interrelationships are completely defined.

LOUs and their relationships to the other parts of the architecture and the outside world can be graphically depicted (see Figure 4-5). This is just another view of the basic relationships that were outlined in the target architecture modeling framework earlier in this section as the “Mother of all Models.”

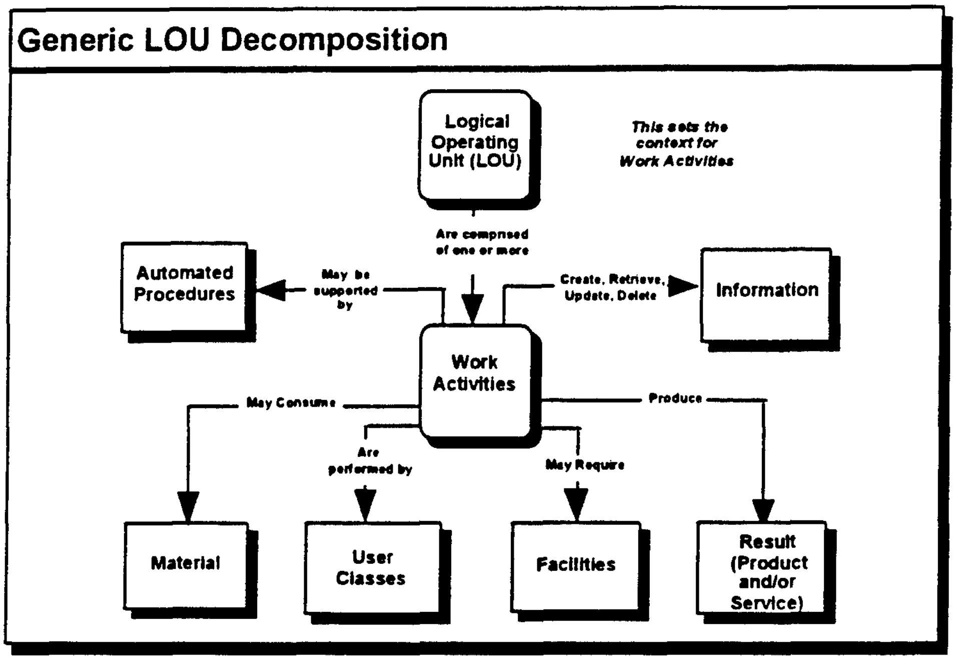
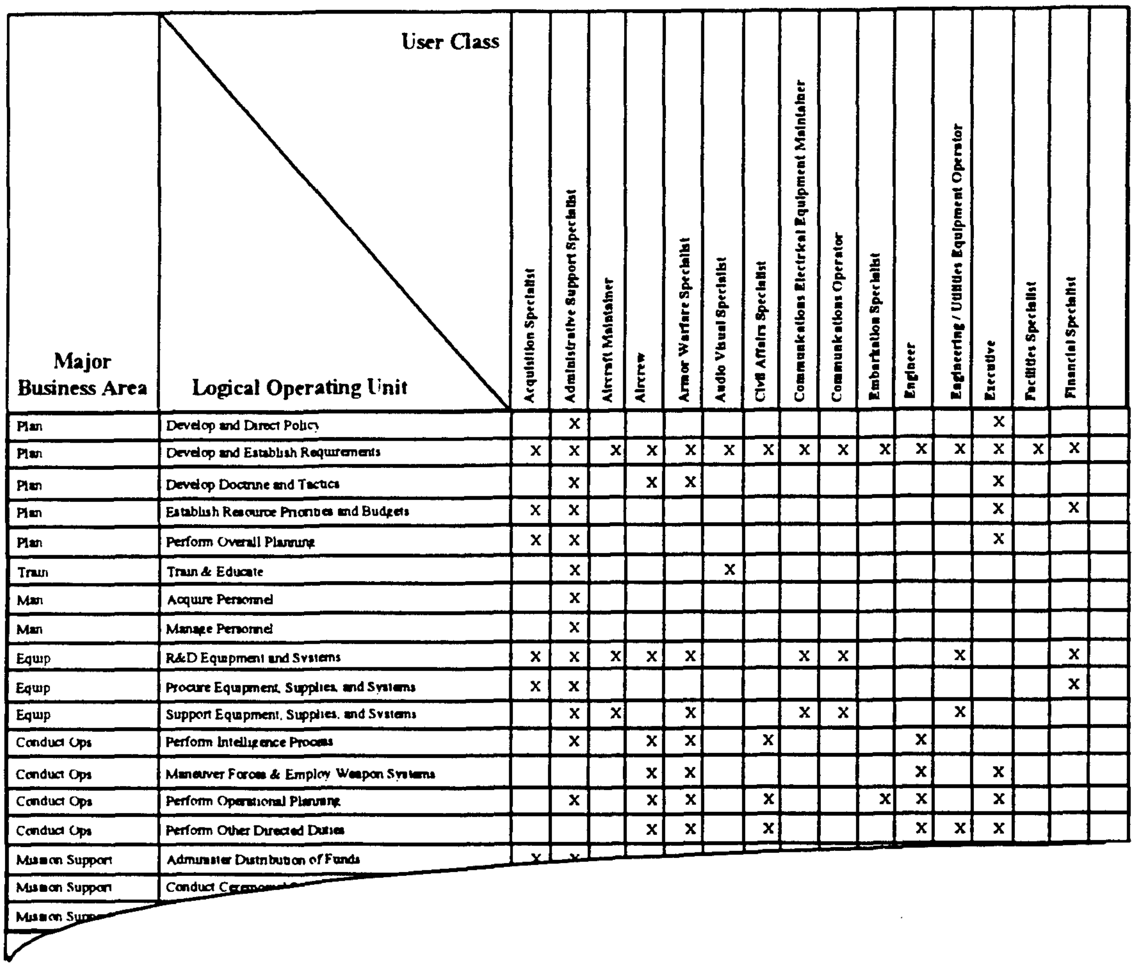

Generic LOU Decomposition
Logical Operating Unit (LOU)
This sets the context for Work Activities
Are comprised of one or more
Automated Procedures
May be supported by
Create, Retrieve, Update, Delete
Information
Work Activities
May Consume
Produce
Are performed by
May Require
Material
User Classes
Facilities
Result (Product and/or Service)

Figure 4-5. Generic LOU Decomposition

As an example of how to use the work view of architecture for analysis, Figure 4-6, the LOU to User Class Affinity Matrix, shows which user classes are likely to perform one or more of the work activities that make up a given LOU. This matrix is a key tool in the analysis of opportunities for automation and the linkage of these automated systems to work locations where these various user classes will perform their work.

Information architecture

The information architecture is composed of high-level subjects that represent all of the information needed to perform the work of the enterprise. The information architecture concentrates on the data being managed in support of the LOUs of work. Each major collection of data needed to support identified functions should be captured in the information architecture.



Major Business Area	Logical Operating Unit \ User Class	Acquisition Specialist	Administrative Support Specialist	Aircraft Maintainer	Aircrew	Armor Warfare Specialist	Audio Visual Specialist	Civil Affairs Specialist	Communications Electrical Equipment Maintainer	Communications Operator	Embarkation Specialist	Engineer	Engineering / Utilities Equipment Operator	Executive	Facilities Specialist	Financial Specialist	
Plan	Develop and Direct Policy		X											X			
Plan	Develop and Establish Requirements	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Plan	Develop Doctrine and Tactics		X		X	X								X			
Plan	Establish Resource Priorities and Budgets	X	X											X		X	
Plan	Perform Overall Planning	X	X											X			
Train	Train & Educate		X				X										
Man	Acquire Personnel		X														
Man	Manage Personnel		X														
Equip	R&D Equipment and Systems	X	X	X	X	X			X	X			X			X	
Equip	Procure Equipment, Supplies, and Systems	X	X													X	
Equip	Support Equipment, Supplies, and Systems		X	X		X			X	X			X				
Conduct Ops	Perform Intelligence Process		X		X	X		X				X					
Conduct Ops	Maneuver Forces & Employ Weapon Systems				X	X						X		X			
Conduct Ops	Perform Operational Planning		X		X	X		X			X	X		X			
Conduct Ops	Perform Other Directed Duties				X	X		X				X	X	X			
Mission Support	Administer Distribution of Funds	X	X														
Mission Support	Conduct Ceremo																
Mission Sup																	

Figure 4-6. LOU by User Class Affinity Matrix

The information view, illustrated in Figure 4-7 is linked to the LOUs identified earlier, showing where the information is created, used, modified, and/or deleted, over time. The information architecture includes a discussion of the principles of information management as well. The AWG makes decisions that should facilitate this information management process. The models should reflect the workshop participant's best judgment about the future uses and characteristics of information within the enterprise. User access to this information across various business locations is also considered here.

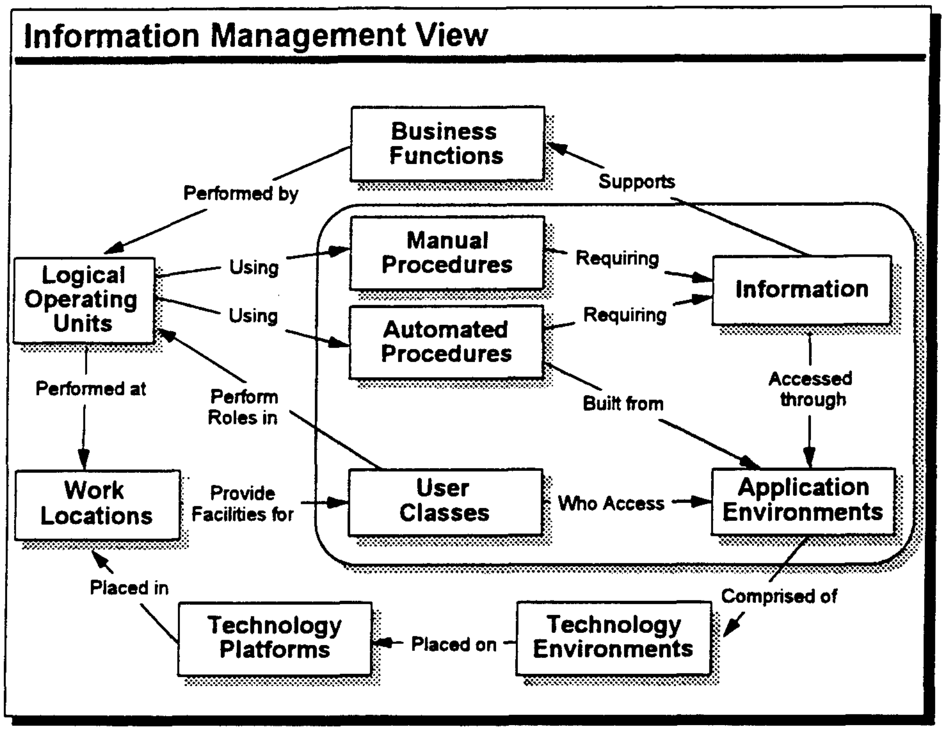


Figure 4-7. Information View of the Architecture

The information architecture for the enterprise will contain three levels of detail, subject areas, data groups, and data attributes.

The LOU to Data Grouping Matrix cross references all of the data groupings defined in the information architecture. This establishes the interrelationships among the data and the LOUs needed to perform the work of the enterprise. It will subsequently be used by systems designers as they develop the projects presented in the applications architecture.

The LOU to Data Matrix, illustrated in Figure 4-8, is used to show which of the LOUs either create, read only, update, or delete data within a given data group. Such a matrix is sometimes referred to as a "CRUD" matrix. This is due to the appearance of the letters C, R, U, and/or D in the cells of the matrix to show respectively Create, Read, Update, and Delete capability by a given LOU. This matrix is used in discussions of opportunities for automation. It is also very useful in decisions regarding the physical location of application systems and the data itself.

Major Business Area	Subject Data Grouping Logical Operating Unit	Agreement	Facilities		Financial	
		Contract/ Agreement	Roads	Structure	Budget	Disbursements & Receivables
Plan	Develop & Direct Policy					
Plan	Develop & Establish Requirements	R			R	
Plan	Develop Doctrine & Tactics					
Plan	Establish Resource Priority & Budget				CRUD	R
Plan	Perform Overall Planning	R	R	R	R	
Train	Train & Educate	R				
Man	Acquire Personnel					
Man	Manage Personnel					CRUD
Equip	Procure Equipment, Supplies, and Systems	CRUD		R	R	R
Equip	R & D Equipment and Systems	CRUD	R	R	R	R
Equip	Support Equipment, Supplies, and Systems			R	R	R
Conduct Operations	Perform Intelligence Process	R	CRUD	CRUD		
Conduct Operations	Maneuver Forces & Employ Weapons Systems		R	R		
Conduct Operations	Perform Operational Planning	R	R	R		
Conduct Operations	Perform Other Directed Duties					
Mission Support						
M						

Figure 4-8. LOU by Data Matrix

Applications architecture

This view of architecture focuses on the opportunities to automate aspects of work and/or the access to information needed to perform work (i.e., the target application systems to support the business). (See Figure 4-9.) Using the work view and the information required by each unit of work within this view, the team identifies application system opportunities, or clusters of functionality, required to support specific business needs. The application view of architecture shows the information usage and flow. The architecture defines the high-level scope and interfaces among applications, not the detailed requirements of each.

The team should identify all future applications that will be needed to manipulate the information and support the work being performed. In the process, the AWG should develop a set of high-level application descriptions. These descriptions are intended to serve as a first-cut view of the major applications.

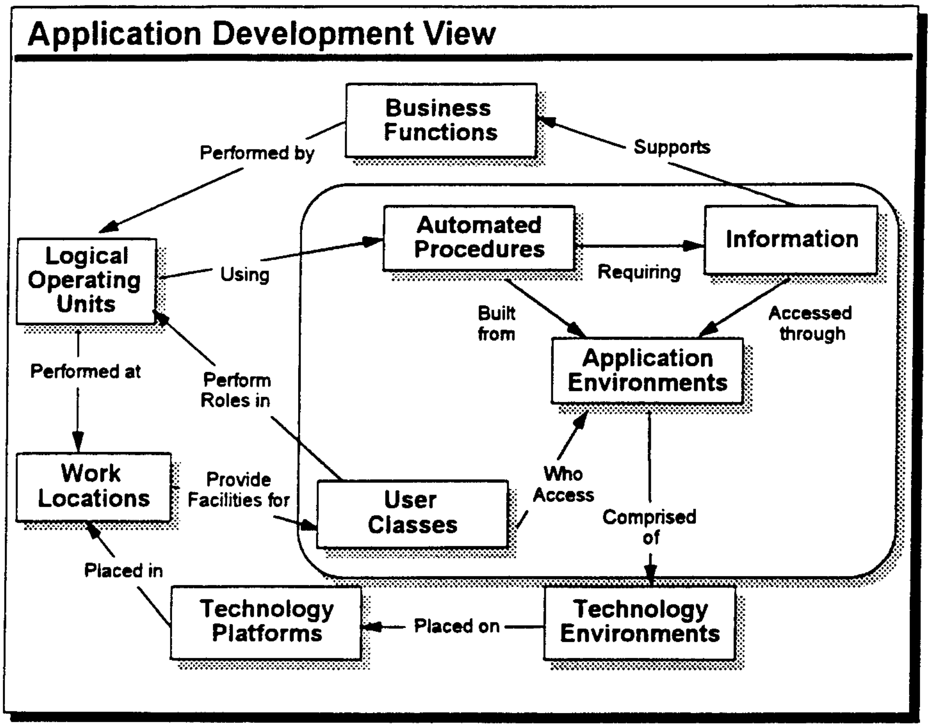
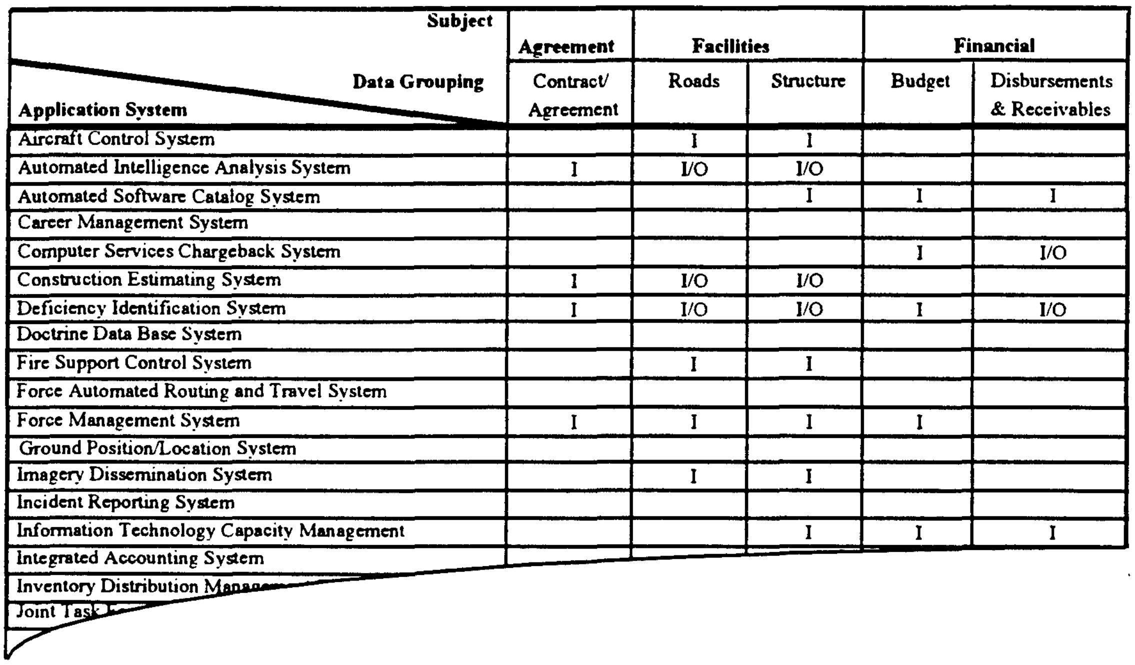


Figure 4-9. An Applications View of the Architecture

A matrix should be developed that shows which applications require read-only access to specific data and which applications may both read and update specific data. (See Figure 4-10 for an example.) Such a matrix is sometimes referred to as an "I/O" matrix. This is due to the appearance of the letters I or I/O in the cells of the matrix to show respectively Input only or both Input and Output capability against particular data. This mapping will be useful in decisions regarding the physical location of the application systems and the information itself.

Technology architecture

This part of architecture development typically requires a reversal of the workshop backroom sessions approach used in developing other views of architecture. (See Figure 4-11.) It is in this phase where, as the old joke goes, "a miracle happens." Usually, the technology architecture models begin to emerge in the mind of a single technology architect who has some quiet time to mull over all of the deliverables of all prior phases and the three views of the target architecture that have already been developed in this phase. This person will have some rules of thumb and years of experience to guide him or her, but it is still somewhat more art than science. This section gives an overview of the thought process that such a technology architect might follow.



Subject / Data Grouping / Application System	Agreement	Facilities		Financial	
	Contract/ Agreement	Roads	Structure	Budget	Disbursements & Receivables
Aircraft Control System		I	I		
Automated Intelligence Analysis System	I	I/O	I/O		
Automated Software Catalog System			I	I	I
Career Management System					
Computer Services Chargeback System				I	I/O
Construction Estimating System	I	I/O	I/O		
Deficiency Identification System	I	I/O	I/O	I	I/O
Doctrine Data Base System					
Fire Support Control System		I	I		
Force Automated Routing and Travel System					
Force Management System	I	I	I	I	
Ground Position/Location System					
Imagery Dissemination System		I	I		
Incident Reporting System					
Information Technology Capacity Management			I	I	I
Integrated Accounting System					
Inventory Distribution Mana					
Joint Tas					

Figure 4-10. Information by Application Matrix

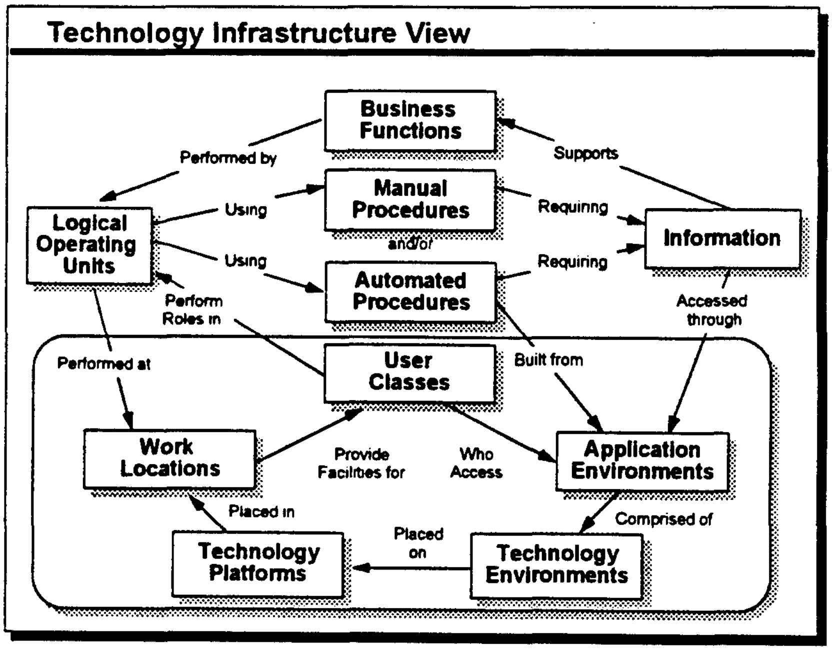


Figure 4-11. A Technology View of the Architecture

This area of architecture uses specific component-level models to provide the basis for linking the technology view of architecture to the work, information, and application views. The linchpin to the other views of architecture is the generic application environment.

With each application area characterized in terms of its generic application environment(s), the other components of the technology architecture can be defined precisely. Using additional component models and generic terminology, the technology architecture will describe the IT infrastructure (framework) required to support the enterprise's objectives as characterized by the other three views of architecture discussed earlier.

Technology architecture building blocks

Three types of building block models (sometimes referred to as constructs) are used in building the overall Technology Architecture Model. They are described below.

Generic application environments

Generic application environments (GAEs) describe types of IT applications and tools needed to support specific application systems. This is the primary building block in linking application systems back to the technology environment.

Generic technology environments

Generic technology environments (GTEs) describe types of services required to support GAEs (i.e., system software). GTEs provide a means of defining a technology environment that has a standard set of characteristics and attributes. Each GTE uses a set of "servers" that provide specific technical capabilities for the GTE. Like the GTEs, the servers are generic components with standard interfaces to the "clients." They are built on, but independent of, specific technology implementations. The result is a layered technology that, if implemented through a rigorously defined set of interfaces, can isolate applications and major technology components from differences in the underlying technology implementation.

GTEs provide the SBA link from GAEs to the technology components and technology implementations within an organization's infrastructure. Each GAE is supported by one or more GTEs. The combination of the GAEs and GTEs provides the infrastructure components for delivering systems and services to the organization.

Generic technology platforms

Generic technology platforms (GTPs) describe the delivery components required to run the applications that ride on the GAEs (i.e., "system hardware").

These generic modeling constructs are planning tools which provide a framework for comparing current and target environments. They also support standards-based architecture planning. They are not, in and of themselves, the final deliverable but are used as a tool to aid in developing the specific technology architecture.

Six technology constructs or GTPs provide fundamental building blocks in a standards-based architecture. Each GTP can function as a fully independent "architecture" in that each has an interface along with processing, storage, and communications capabilities. As such, each GTP may offer alternative choices in delivery of the same GAE. For example, all six constructs are capable of supporting some form of electronic mail, with different associated strengths and weaknesses. These six GTPs include:

- Intelligent WAN systems
- Establishment-based switching systems
- LAN systems
- Enterprise or corporate processing systems
- Divisional or departmental processing systems
- Desktop or portable intelligent workstations.

It is important to note that the GTPs do not connote a particular size/capacity. The names for the GTPs connote the usage of the processor, not size. In fact, departmental processors may be larger or smaller than enterprise processors. Some processors acting as LAN servers could well be larger than departmental or enterprise processors depending on the way a given organization wishes to organize its work.

How to use the building blocks

The generic building blocks just described are useful in the process of developing the target technology architecture. The end result of such a process is best shown by the use of an example. The target technology architecture developed for the U.S. Marine Corps provides an excellent example of the output from this process, and the reader is referred to the target architecture deliverable from that project.

The thought process that was used to produce the USMC technology architecture is guided by technology "rules of thumb" based on experience and informed by the other views of the architecture. Specific characteristics of work, information, and applications enter into the interpretation of these rules. Some of these rules are:

- Keep the processor as close as possible to the users of systems residing on the processor
- Maximize independence between major application groupings (stepwise escalation from loose coupling to tighter coupling)
- Within major groups of applications, look for ways to gain tighter coupling (such as shared databases)
- Establish the smallest practical set of standards as possible
- Maintain vendor independence in standards for as long as possible
- Take locations into account but do not "agonize." (Follow accepted rules of the road and the effect of being "off" on locations will be minimized.)
- Be pragmatic—do not wait for the ultimate environment. Build up to it by accepting some short-term compromises while keeping as many options open as possible.

In addition to this guidance, there are other practical issues to consider about the placement of applications on technology platforms. The support requirements of the applications can be used to assess which platform is a best candidate for placement. For example, highly individualistic applications and tools (e.g., text processors, CAD/CAM, CASE) have a high affinity for the desktop. Applications requiring the need for terminal support or which act as the server side of client/server applications have a high affinity for departmental or enterprise processors. Finally, infrastructure applications such as E-mail, EDI, and other common services have a high affinity for LAN or other network servers.

Techniques to arrive at the target technology architecture

There are recommended steps to follow to analyze the other architecture views that will facilitate the process of defining the target technology architecture.

First, begin by reviewing the characteristics of information. Produce a first cut map of the technology platform using rules of thumb. Then, review the characteristics of applications. This should result in a first cut map of each application to technology platform where the bulk of the most demanding data resides.

The CRUD matrix should next be reviewed to gain insights about potential data sharing and the effect this will have on data distribution. Also, the application to information (I/O) matrix should be reviewed for similar insights (and potential adjustments).

Each of these steps is performed in an iterative fashion until all applications, data, and associated platforms are mapped to logical work locations. By now, a reasonable model should begin to emerge that can be tweaked by looking at the form of information and the potential impact on network traffic. Finally, with all of these steps complete, some judgment calls can be made about the style of computing:

- Distributed presentation
- Remote presentation
- Distributed function
- Remote data management
- Distributed data management.

Capacity requirements should be considered as well to finalize the model. This last step represents the final "proof" of the model. The information volume, timeliness, and currency requirements, along with application availability and reliability, can be used to make a guess at the scale of the processing platform required at each location. The volume, timeliness, and currency requirements can be used to categorize the network transmission capacity needed between locations. The result of this examination of capacity issues may cause some final adjustments in application and information distribution across the network.

Standards model

To implement the standards-based infrastructure, it is important to consider the scope and depth of the standards to be adopted. Fundamentally, all cases of standards adoption require answering three questions:

- What standards should I adopt?
- Where in my architecture should I adopt them?
- When should I adopt them?

Both TAFIM Volumes 2 and 7 should be used in this phase. Volume 2 suggests a standards-based model for user interface, database, applications, operating system, communications, languages, management, and other services. Volume 7 identifies the standards and specifications approved by DoD as the method for satisfying those service areas. Architects are encouraged to select appropriate standards and specifications from Volume 7 to form a standards profile. Profiles vary as functional requirements vary. The AWG must be prepared to define the details that underpin each section of the diagram for their functional area's particular implementation. Appendix C on detailing the target architecture can also be a good reference point for teams attempting to define the details of the standards model.

De jure vs. de facto standards

A target architecture must be developed such that it will permit implementation migration towards full standards compliance — described as either de jure or de facto.

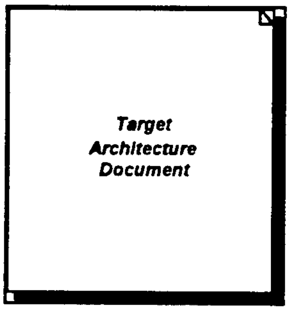
Business requirements should not be compromised strictly for the sake of "open systems." However, whenever a de jure standard is available in effective price/performance product form, it should be implemented as quickly as possible. Specifically, the de jure standards should be:

- Specified in policies, guidelines, and architecture

- Products and services based on standards-based policies and guidelines selected whenever viable competitive cost/performance alternatives to proprietary solutions exist in the marketplace.

Product implementations today tend to be based more on de facto standards than de jure standards. The target architecture effort should take this reality into account. Products based on de facto standards are more available today in the marketplace. A standards-based architecture based solely on de jure standards may be elegant and pristine in concept but can also be essentially sterile because so few of the adopted standards are actually in the marketplace via vendor implementations. All effective standards-based architectures must acknowledge the hybrid nature of this reality.

Creating and publishing the deliverable



The target architecture is one of the more creative aspects of the SBA process. The deliverable is arrived at only after significant thought has been invested in an iterative review of the baseline material. The architecture principles should be clearly reflected in the target architecture, and the technology view should be capable of supporting the new work processes envisioned in the target.

- Clarity of Target Architecture Document
- Management acceptance of Target Architecture Document.

Effectiveness measures

- Ability to map from current embedded base to target architecture
- Inherent flexibility in the SBA action plans.

The effectiveness of the *Target Architecture Document* will ultimately be determined by the degree to which it is used by the DoD. As discussed earlier, the document must be easy to understand and must set a reasonable target, otherwise no one will use it.

Technology and tools required

- Word processing tools (with graphics capabilities)
- Spreadsheet tools
- Business graphics and drawing tools
- Work room for AWG meetings.

It's important that the blueprint document be highly visual (i.e., contain many diagrams and checklists). The easier the document is to understand, the more likely it is to be used and referenced.

Appropriate resources should be dedicated to creating a user-friendly blueprint. Some organizations have even gone so far as to hire layout artists to streamline the document. While this kind of zeal is not required, too much emphasis cannot be placed on making the document easy to use (i.e., technology should be available to facilitate the creation of a quality deliverable).

Staffing skills required

- Experienced planners
- Business professionals
- Acquisition experts
- Information technologists
- Writing and presentation skills.

The ASC will provide guidance, direction, and high-level review for the work of the AWG. The AWG will be responsible for assisting in facilitating working sessions and for producing the deliverables in the planning process. This team will have broad, non-overlapping backgrounds in the business to be modeled.

Key executive and knowledge workers need to be available for interviews and/or workshop sessions according to a schedule to be developed within the initial weeks of the project.

The AWG will develop a working *Target Architecture Document*. This document will then be reviewed with the ASC and other key stakeholders within the enterprise (see Figure 4-12).

The committee then sponsors a draft document that is reviewed, amended, and approved by the appropriate players within the enterprise. This is typically a management group composed of functional area heads and the Chief Information Officer or his/her equivalent. In some organizations, the chief executive will review the SBA document.

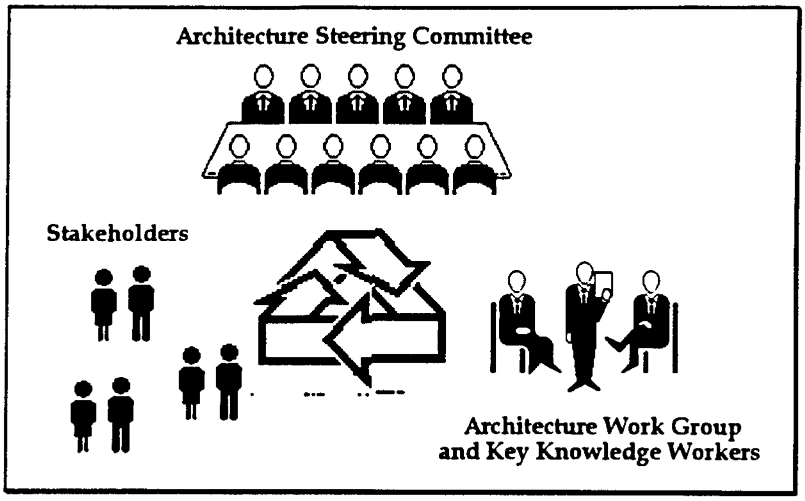


Figure 4-12. The Review and Approval Cycle

Completion criteria

- Creation of reviewed and reconciled models
- Creation of target standards (if part of agreed-upon scope)
- Management acceptance of target definition
- Acceptance of *Target Architecture Document.*

Standards, as articulated in the policies and guidelines section of this document, will be the core to enabling construction of the standards-based infrastructure. This document will be a key input document to the remaining steps in the implementation cycle, particularly in identifying opportunities and migration options.

Issues

- Workload of architecture work team(s)
- Target architecture scope management
- Key knowledge workers' availability for workshops
- Trained, experienced standards-based architects
- Correct understanding and anticipation of the future.

It is essential that the AWG be properly trained in SBA planning practices and that members be full-time participants in the effort. This implies that management eliminate the pro forma activities that team members are typically required to perform.

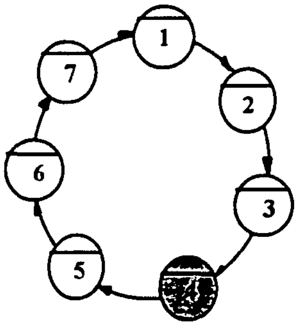
Failure to make a commitment to this effort seriously can result in the execution of another tired planning exercise that carries little or no weight within the function after its completion. The old adage “you get out what you put in” truly applies to SBA planning projects.

Section Five: Opportunity Identification

Contents **Page**

Figures

Section description



This section describes the overall process by which the AWG categorizes and identifies opportunities for exploiting the target architecture. Opportunity identification is the phase dedicated to identifying the projects needed to move the organization from the present to the future (the target architecture). This phase defines the parameters of change, the major steps along the way, and the major activities to be undertaken.

Objectives

To identify key opportunities for implementing the target architecture environment on a "fast path" basis while also developing a context for development of migration options and detailed implementation plans.

In the opportunity identification phase, projects necessary to move the organization from its current environment (as defined in the baseline deliverable) to its target environment are identified. This includes a detailed description of the automated and non-automated initiatives that will be necessary to reach the target architecture. This phase will flesh out the application, non-application (technology infrastructure), and non-technology initiatives that should be implemented to achieve the vision of the organization.

At this stage in the project, it is sufficient to provide documentation of the essential steps needed to achieve the target and not to provide a cost/benefit justification for these projects. This will be done in the migration options phase as projects are justified and ordered into plateaus, and dependencies between projects are identified.

The AWG identifies various opportunities through workshops and work group analysis. These opportunities are tested and filtered by the business and IT functions. Once finalized, the opportunities are documented in the *Opportunity Identification Document.*

Scope

Throughout the AWG process, numerous opportunities are identified for introducing standards-based architectures and harvesting benefits associated with the proposed architecture solutions. During this phase, the identified opportunities are classified with regard to a number of criteria. This classification scheme becomes the foundation for migration planning and implementation.

Experience shows that if the project cannot deliver fundamental opportunities on a short-term payoff basis within 3 to 6 months, the rest of the standards-based architecture will probably never be implemented. Results are critical to success. This places a premium upon identifying opportunities that are:

- Short and medium term in nature
- Low risk, high payoff in implementation
- Offer a high degree of freedom within the existing architecture so that they may be implemented easily and migrated to as quickly as possible.

As is customary, many opportunities are identified at the same time that the application component of the target architecture is being developed. Therefore, the systems introduced there appear here as project opportunities. Also included is the definition of infrastructure projects (i.e., technology features that must exist in order for the applications to run) and non-technology projects (i.e., non-systems projects that are necessary to achieve the vision of the future presented in the target architecture).

Figure 5-1 illustrates the contrast between the traditional information plan and the standards-based fast path implementation focus:

Deliverables

An *Opportunity Identification Document* that contains:

- Description of project opportunities
- Dependencies
- Issues to be considered.

In this phase, the team describes the opportunities in general terms, the size and scope of opportunities, as well as the dependencies that need to be considered when the time comes to deliver the project. A sample outline for this document is included in Appendix I.

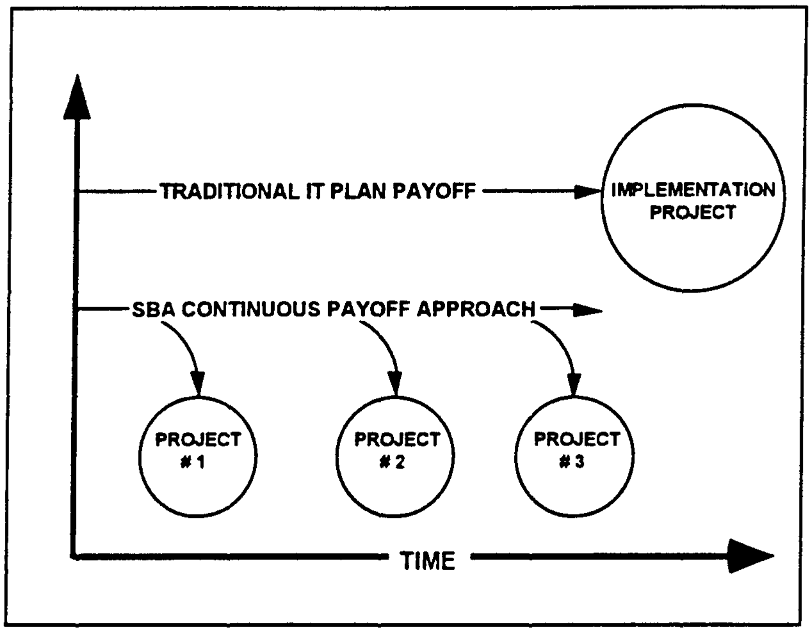


Figure 5-1. Implementation Payoff Approaches

Critical success factors

- Understanding of the implementation challenges and payoffs at a high level
- Understanding of the Baseline Characterization Document, Target Architecture Document, and other source data
- Experience in business and IT planning
- Practical understanding of the tradeoffs between business issues, technology, tactical, and operations settings
- Understanding of Federal procurement guidelines and issues
- Working knowledge of systems development and maintenance
- Familiarity with IT security planning
- A systems migration planning background
- An effective communications vehicle between team members and from the ASC and the AWG.

In this phase, it is important that the AWG balance the strategic long-term objectives of the target architecture with a reality-based tactical view of what may be accomplished in the near- to mid-term time frame. Grand plans are indeed grand; in most cases, they either fail or never see the light of day. Unfortunately, most implementation efforts are judged by the first projects delivered rather than on the merits of the overall design rationale. Thus, the demonstrable practicality, efficiency, and effectiveness of the proposed projects will be used to assess the success of this effort.

Constraints

- Working within the current architecture paradigm may limit the team's ability to "see" new opportunities
- Vision (or lack thereof) may limit successful execution
- Lack of a coherent business case.

Many times, implementation efforts focus only on tactical programs. The ability to discern opportunities is only increased when team members have a structured approach and are able to see beyond the constraints of the current environment.

Task list

- Initiate task
- Identify gaps between baseline and target architectures
- Identify payoff categories
- Identify key payoff projects
- Draft *Opportunity Identification Document*
- Conduct review with ASC
- Finalize *Opportunity Identification Document*
- Distribute *Opportunity Identification Document.*

Gap analysis

Determine the "gaps" between the baseline and the target in all four views of the architecture. Spreadsheets are a good tool for this. One approach might be: across the top, list all of the "target" components of a given view (e.g., future business processes, future information facets, future applications, or future technology components). Along the left-hand column, list the current components. In each cell of the spreadsheet, account for all current components. Some current components may be eliminated. For example, an "auditing" work process may be "non-value added" for the future; therefore, it is eliminated. For cases such as this, create another column in the spreadsheet entitled "eliminated." On the other hand, new components may be added. For example, a new service process may result in higher satisfaction for the user of the service. For cases such as this, create another row entitled "new." All eliminated components and new components create gaps. The identification of opportunities must fill these gaps. Figure 5-2 below illustrates this technique for determining gaps.

Target Current	Solicit Business	Fill Order	Provide Customer Service	(Eliminated)
Take Order				GAP
Fill Order		Okay		
Audit				GAP
(New)	GAP		GAP	

Figure 5-2. Gaps Between Baseline and Target Architectures

Payoff categories: the opportunity context

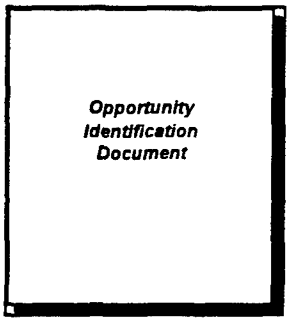
A number of benefits are associated with open systems and standards-based architectures. The TAFIM series highlights the implementation opportunity initiatives that support portability, scalability, and interoperability of applications and systems. As such, it defines an "opportunity vision" for the future. It was devised to permit the DoD to take advantage of the benefits of open systems and new standards-based technologies available in the commercial market.

Specific objectives for the DoD TAFIM include:

- Improving user productivity
- Improving development efficiency
- Improving portability and scalability
- Improving interoperability
- Promoting vendor independence
- Reducing life-cycle costs.

These objectives may be used as categories for evaluating implementation "payoff" opportunities (see the TAFIM, Volume 2 for more detail.)

Creating and publishing the deliverable



The key deliverable in this phase is the *Opportunity Identification Document.* It should focus on providing the ASC with a high-level understanding of the opportunities at hand. As described in the opening of this section, the document should focus on highly visible short-term payoff projects with a "continuous payoff" approach to implementation opportunity identification. The document's entire objective is to describe the nature of the target architecture opportunities and the role they will play in closing the gap between the baseline environment and the target architecture.

Effectiveness measures

- Degree to which implementation plans can be developed
- Management enthusiasm regarding opportunities identified.

This phase will vary widely in terms of calendar time required for completion based on organizational culture, individual schedules, and the formats that organizations are accustomed to using. Ideally, when conducted on an intensive basis, this phase may be completed in 6 to 10 weeks. The draft and final iterations of this document should be reviewed with the ASC before any action is taken and changes made accordingly. As with other deliverables, the document should go through several draft iterations before being approved by the ASC.

Tools required

- Word processing and graphic presentation packages
- Architecture team room for meeting
- Spreadsheet tools and/or user-friendly personal computer-based database packages for inventory logging.

It is key that the AWG put together a high-level presentation for the ASC that highlights the features and logic of the implementation opportunities it has identified. "Selling" the architecture to the ASC must be done on this basis.

Staffing skills required

- Migration planning skills
- Software modeling skills
- Writing and presentation skills
- Organizational data collection knowledge
- Familiarity with word processing, presentation, spreadsheet, and database packages that run on most popular personal computers.

This phase requires individuals who are familiar with project definition and who understand the requirements of the next phase in the process, which will assess the benefits and risks associated with such projects as well as the priority which should be placed on each. Ultimately, each of the projects must be justified in terms of its contribution to the target architecture or as a stand-alone project. The goal of this step in the process is not to encourage the creation of an undisciplined wish list. Rather, there is every expectation that the minimum set of projects (automated and non-automated) necessary to achieve the vision will have been identified.

Completion criteria

- *Opportunity Identification Document* completed
- Management acceptance of *Opportunity Identification Document.*

This phase is completed when the ASC accepts and signs off on the *Opportunity Identification Document.* It is important that all the ASC members, as well as the AWG, have a shared understanding of its content since it will become the basis for developing migration options and for implementation planning.

The AWG should obtain a sign-off that ensures full ASC approval as with all other steps in the process.

Issues

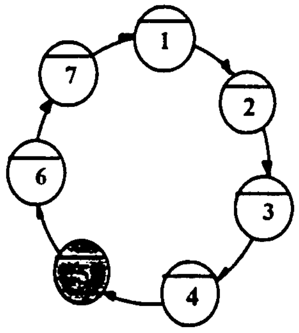
- Executive "buy-in"
- Workload of work team(s)
- Consulting required
- Training required
- Subject matter expert availability.

Section Six: Migration Options

Contents **Page**

Figures

Section description



This section describes the overall process by which the AWG identifies and develops migration options for moving to the new target architecture. This section also describes the overall process by which the AWG categorizes and identifies opportunities for exploiting the target architecture and shows how such opportunities can be justified in areas such as their cost-to-benefit ratios or the role they play in providing support for future projects to be implemented as part of the target architecture. Included in this activity are descriptions of how the *Migration Options Document* is developed.

Migration planning is the phase in the process when all essential projects are sorted into plateaus for implementation planning. The sort process is based on the interdependencies between projects. In addition, projects are sorted by strategic value. Those with greatest payoff or strategic significance should be implemented as early as possible to take maximum advantage of the value they represent. Finally, cost is considered in developing the implementation plan. Cost is an important consideration in recognition of the fact that budgets are limited and most, if not all, expenditures must be justified in terms of the benefits they will provide or in terms of the essential infrastructure support they represent. The following provides a feel for the content of this phase:

- Estimates of the work and resources required to migrate from the current environment to the target environment are developed with resource estimates and responsibility assignment.
- Comparison of target to baseline architecture is performed to identify areas where the current situation satisfies the target requirements and where gaps exist.
- High-level plans for migrating from the current to the target architecture are described and dimensioned.
- The migration plan must account for organizational change and must also be flexible enough to accommodate changes in the architecture itself as the migration plan is being implemented. We refer to this last step as "innovation-proofing" the architecture. The

output from this phase is similar in nature to the document that is produced after the architect has blueprinted the architecture of a building project—a "construction plan" that tells the builder how to actually erect the building.

Objectives

To develop a comprehensive, prioritized set of project initiatives, which, when completed, will move the enterprise from the current state to the target architecture.

The AWG identifies alternative construction options. Major critical implementation steps are developed by the AWG. The detailed implementation plan is then reviewed, not only with IT, but with functional area personnel to assure that time frames are realistic and goals achievable. Project implementation responsibilities are assigned, as well as implementation dates, based entirely on functional area requirements. This entire phase is documented by the AWG in the *Migration Options Document.*

Scope

This phase will identify all projects required to fully implement the target architecture.

The AWG must determine how many areas of the target architecture to tackle at one time as well as the interdependencies between the components. Theoretically, all four views of the target could be pursued simultaneously. However, practically speaking, they will be easier to manage if they are handled in an independent but related manner. These two conceptual approaches are shown in Figure 6-1.

After a high-level determination is made on which of these dimensions of the architecture are to be addressed, and a high level description of the necessary projects has been created, the scope of each project is defined. This should include a project statement, a scope definition, the major components of the project, and major steps to be covered during the project's life cycle.

A project scope statement addresses and delimits a project that is as small as putting a standard user interface across a group of applications. Alternatively, the project could be on a much larger scale wherein all major work processes in a customer service environment, as well as the standards-based technology to support them, are reengineered.

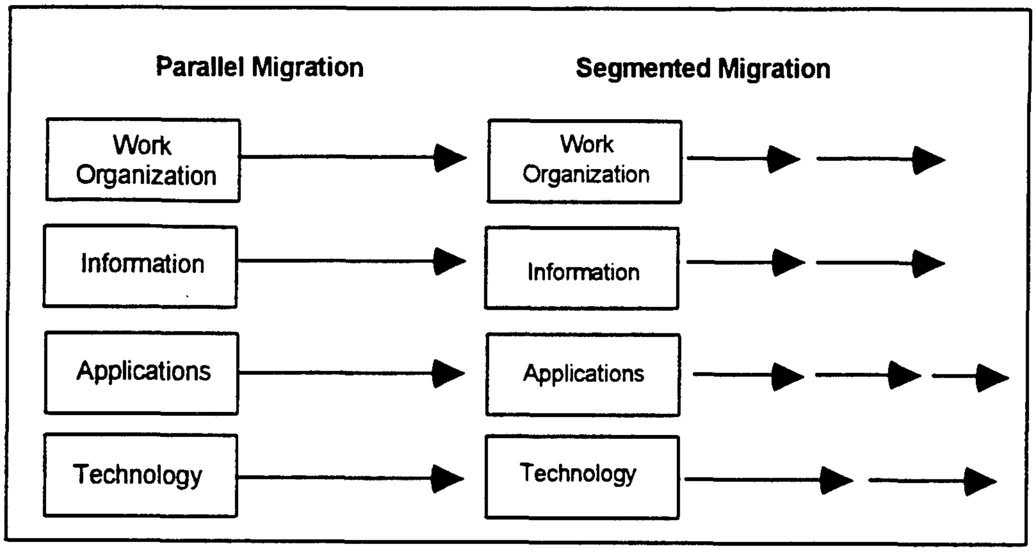


Figure 6-1. Migration Approaches

Deliverables

The *Migration Options Document* provides specific recommendations for the priority of the project initiatives that must be performed to move the enterprise toward the target architecture. This document should include a thorough discussion of the migration plateaus. A sample outline for this document is included in Appendix I.

Plateaus are fashioned to deliver "clusters" of business benefit. There are usually three plateaus, with the first plateau containing some "quick hit" projects as well as the highest priority major projects:

- Plateau 1 — projects beginning in years one and two
- Plateau 2 — projects beginning in years three and four
- Plateau 3 — projects beginning in years five and beyond.

This document should indicate the priority order of the projects and ballpark costs associated with each plateau. The following are the key sources of information (from prior phases of the SBA) that are used in the migration options phase.

Baseline application assessment charts

These deliverables classify all existing applications as to recommended disposition based on target architecture requirements and the rating of the existing applications against standard criteria. The result is that the applications are placed in one of the following four categories:

- Renovate/reengineer
- Replace/discard
- Keep/tune
- Asset/build upon.

Target application characteristics

This deliverable provides a number of characteristics of envisioned application systems for use in prioritizing these applications. The most important characteristic is the application's perceived contribution to strategic drivers (i.e., a measure of the strategic significance of the target application). This allows the target applications to be sorted in order of highest strategic significance.

Target application to existing application matrix

This deliverable provides the connection between identified future application functionality and existing applications that may currently supply some (or possibly all) of this functionality. It combines this mapping with the assessment of the existing application and the target application's strategic significance.

These source deliverables provide much of the rationale for the prioritization. They are also valuable in arriving at the ballpark cost estimates.

Critical success factors

- Understanding of implementation challenges and payoffs
- Experience in business and IT planning
- General cost/benefit orientation towards technology planning
- A team that has experience in implementing one or more of the target areas (i.e., work flow, application, etc.)
- Migration options that avoid full conversions.

Conversions tend to conflict with functional area priorities. Migration to open systems will take many different paths for users. It will depend upon the embedded base of existing systems and the rate and speed the enterprise seeks to move into target systems over time.

If open systems standards are specified in the target architecture, other considerations must be reviewed. For most organizations, the move into open systems will mean maintaining separate environments over some period of time and running parallel environments. This should be factored into the business case for open systems. When the overall case is examined in terms of long-term benefit, the parallel environment will be most cost effective. This is typically the case in spite of the fact that the initial and additional cost of running parallel environments may skew the cost case against parallel facility-based migration.

Delays in implementing a migration strategy to a standards-based architecture may ultimately increase the number and effort of conversions required.

Constraints

- Inexperience in migration planning may limit the team's ability to develop a realistic and acceptable set of migration options.
- The existing work organization may be unable to adjust to the options defined.

As part of the *Migration Options Document,* it is important that the AWG consider issues surrounding organizational change processes. These include, but are not be limited to, the establishment of an ongoing architecture review board and process. The architecture management function itself needs to be authorized to specify architecture standards, administer implementation of the additional strategy, roll out standard tool sets used in the SBA process, and audit compliance with those standards. Thought should be given to establishing a system architect role or function, if the function does not presently exist.

In addition, it would be helpful to describe the various work flow and organizational change processes associated with implementation of the new architecture. This should be an integral part of the overall planning process, because this is where the synergy of organization and standards working together will be most powerful. These and other concepts are covered in more detail in the final phase of the SBA process, SBA administration.

Some of the issues to be faced in this phase are listed below. The AWG should review, modify, and extend this list to make it more meaningful to its specific DoD functional area. This can help ensure success in the SBA process.

- Embedded legacy systems must remain in place for some time for investment or work force resource reasons.
- Open system products which implement de jure standards simply do not exist for many requirements.
- Proprietary solutions can be very effective price/performance solutions if the larger cost savings associated with implementation of open systems are not well understood.
- Organizational inertia—implementing technological change is as much a cultural, organizational, and political challenge as it is a technical process.
- Lack of cohesion between the IT technical community and the function-oriented players.
- Lack of an organizational strategic vision can lead to squandered resources as funds are spent on insignificant or inappropriate efforts.

Lack of a planning and implementation process with which to identify common requirements for standards-based systems.

It is important for the team to remember that with standards-based planning it is possible to eliminate entire classes of technology and replace them with new technology platforms. For instance, an organization can:

- Move applications from mainframes to mid-range platforms
- Move applications from mid-range platforms to high-power networked workstations
- Move applications from master/slave implementations to cooperative processing implementations within an existing proprietary architecture
- Migrate connectivity services (such as E-mail) from proprietary mid-range platforms into a diverse, multiplatform standards environment (X.400) with a parallel strategy for directory services (transition to X.500)

- Implement UNIX-based workstations and servers and replace an entire existing application and platform portfolio.

Task list

This phase will determine the migration plateaus needed to reach the vision by the target date. It is improbable (and probably not recommended) that the organization will want to implement the vision all at once. Usually the vision is attained (or the architecture implemented) by achieving a series of objectives, each of which builds upon the prior, until the vision is attained. The migration plan includes the tasks, timing, dependencies, and resources needed to achieve all the plateaus described in the migration strategy.

- Determine the gaps
- Use any available examples of applicable work
- Determine pace of change desired by the enterprise
- Determine the migration plateaus needed to reach the vision by the target date
- Determine components (work, information, applications, and technology) required to achieve the vision
- Produce migration plan implementation alternatives
- Include security planning migration considerations
- Draft *Migration Options Document*
- Conduct review with ASC
- Finalize and distribute document.

Gap analysis

This process phase is based on the gap analysis between baseline and target architectures. (See Figure 6-2.) The pace of change (i.e., how soon the enterprise wants to complete the implementation of the architecture), along with the priority and interdependence of the projects, will contribute to defining the plateaus needed to accomplish this vision.

Once the target architectures have been developed, the AWG should determine the degree to which the existing technology environments, applications, and platforms support the target environment(s). The data collected during the baseline characterization phase should be useful in this effort.

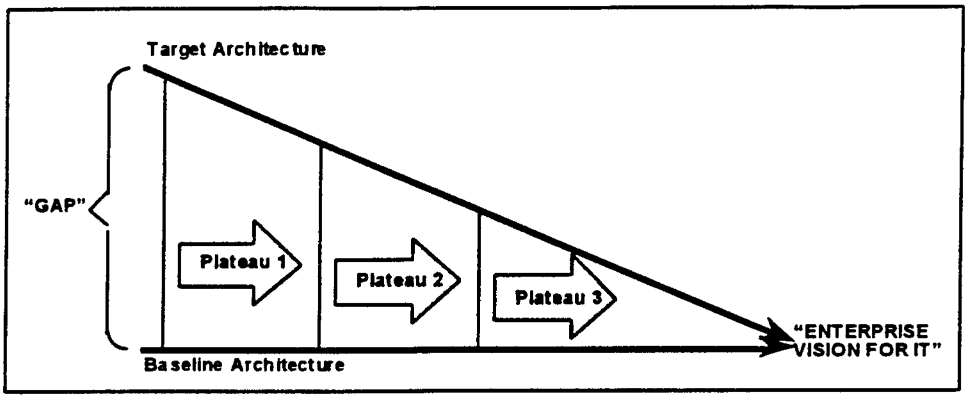


Figure 6-2. Closing the "Gap" Between Baseline and Target Architectures

Opportunity categorization

To initiate this activity, the AWG begins by categorizing each of the opportunities identified during development of the target architecture into three categories:

- Magnitude classification
- Risk classification
- Degrees of freedom classification.

After the opportunity is reviewed in terms of these considerations, the details of these classifications are put into the business cases for implementation consideration.

Magnitude classification

Primarily, the AWG seeks to determine whether or not the opportunities represent major architecture shifts from existing legacy systems in place or an incremental move towards standards over time. The team seeks to classify opportunities in terms of "moves" that may be made in standards-implementation over time.

In Figure 6-3, a user has decided to replace an entire proprietary system with a POSIX-compliant architecture implemented under an X/Windows user interface within a short time interval. Based on the architecture framework, baseline characterization, and target architecture work conducted by the AWG, this solution appears to be quite attractive from every dimension but must be characterized as a "radical" move. Every aspect of the "old system" architecture will be changed in quickly moving to the "new system" architecture.

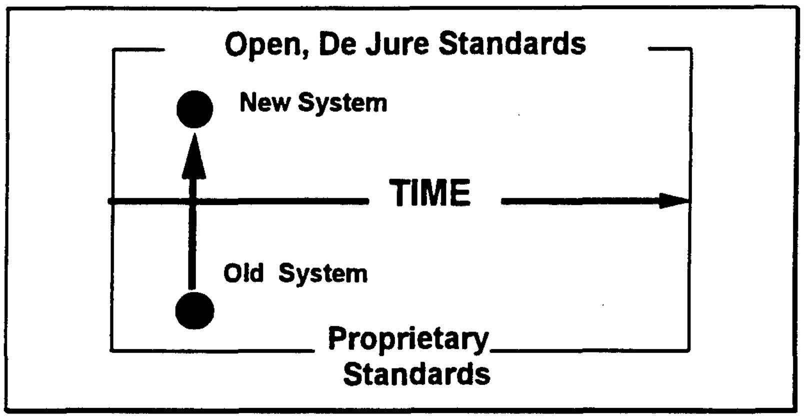


Figure 6-3. Radical Move to Open Standards

In Figure 6-4, the AWG has gone through the same planning process as the one previously described. However, it has decided to implement only OSI connectivity solutions within its proprietary "old system" architecture over the next 3 years. It will adopt SQL whenever possible in its database design activities, but only for new systems. Old databases will remain non-SQL compliant. Other than these two standards-related activities it will remain, for all intents and purposes, proprietary in its "new system" architecture, evolving towards "openness" over time. These moves may be characterized as *incremental.*

Risk classification

In addition to characterizing opportunities as incremental or radical in nature, they may be characterized in terms of risk as shown on the following two matrices. In Figure 6-5 the ideal "low risk, high payoff" opportunity is described in terms of migration.

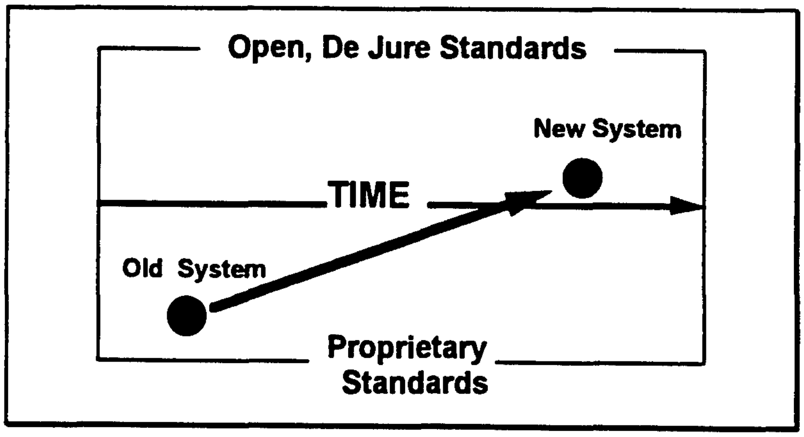


Figure 6-4. Incremental Move to Open Standards

Thus, we may use this example to classify an opportunity where the user is moving from a proprietary SDLC communications protocol to an open protocol such as X.25. In this case, the user is attempting to connect diverse functions via standards internationally. The new system is based on X.25 OSI packet switching protocol. Because X.25 is an established international standard and is widely available in products, it is therefore a low risk move. As a result of its implementation, the two hypothetical international functions will be able to connect their networks together quickly. The opportunity is high payoff in nature.

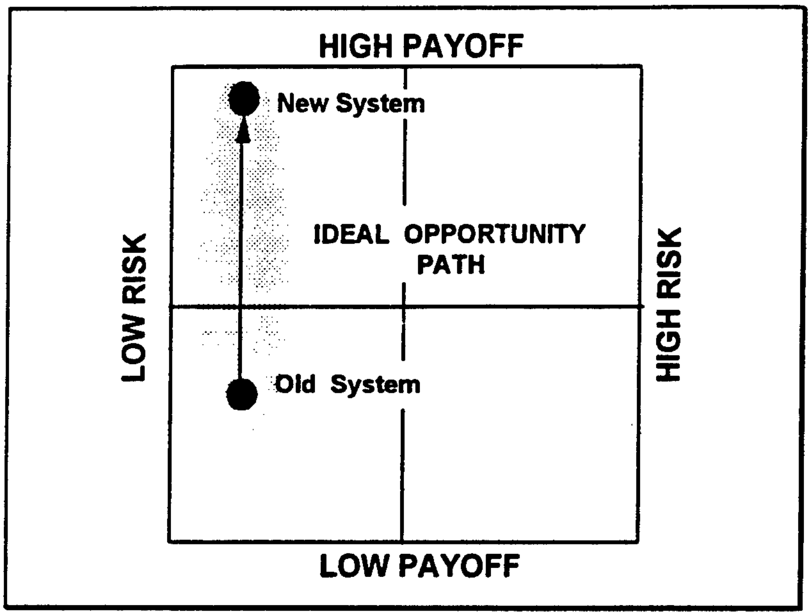


Figure 6-5. Risk: Ideal Migration Path

More often than not, however, the typical IT manager sets out to deliver a "low risk, high payoff" opportunity only to find himself or herself implementing a "high risk, low payoff" solution. Figure 6-6 shows this situation, as contrasted with Figure 6-5 above:

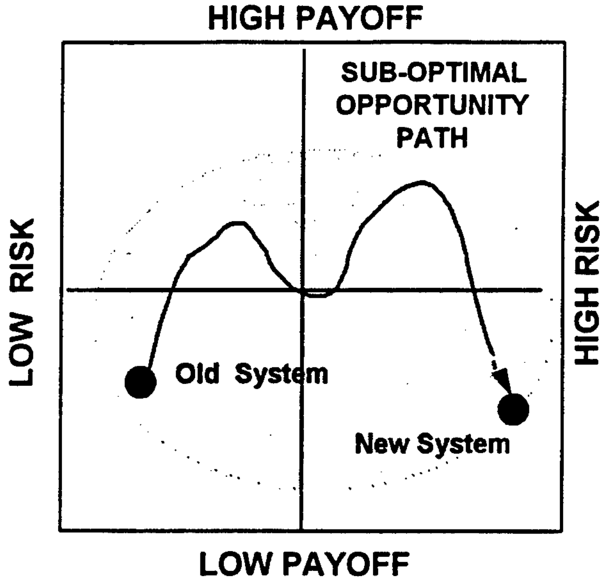


Figure 6-6. Risk: Typical Migration Path

An example of how an IT manager might set out to implement standards and end up with a "high risk, low payoff solution" opportunity may be illustrated with X.500 directory standards.

In this example, a user decides to implement X.500 in a new target system for directory management for the evolving electronic mail application based on diverse LAN environments. Since X.500 standards are not complete, the user assumes the gamble that the X.500 standard will be completed within 48 months and will be widely available in products. In fact, *the standard is fully specified and completed in the user's hypothetical 48-month time frame but is not implemented in products* as quickly as the user requires.

In this imaginary instance, the LAN-based electronic mail users cannot find other electronic mail users on distant LANs throughout the function, because the system was implemented with a key standard architecture component missing. The result is chaos.

Degrees of freedom classification

A third way to conceptualize standards and their implementation and categorization is to describe the opportunity in terms of "degrees of freedom." Degrees of freedom describe the degree to which, given the current architecture, you are free to adopt open-system-based standards and technology and achieve significant benefits in a new architecture in relatively short order.

If the current architecture does not allow you to implement open standards quickly, then you will be consigned to a slow migration (low payoff). On the other hand, if your current architecture permits you to implement open standards quickly, you have a high degree of freedom within your existing architecture, and you will be able to migrate to your new architecture quickly (high payoff). This concept is illustrated in Figure 6-7.

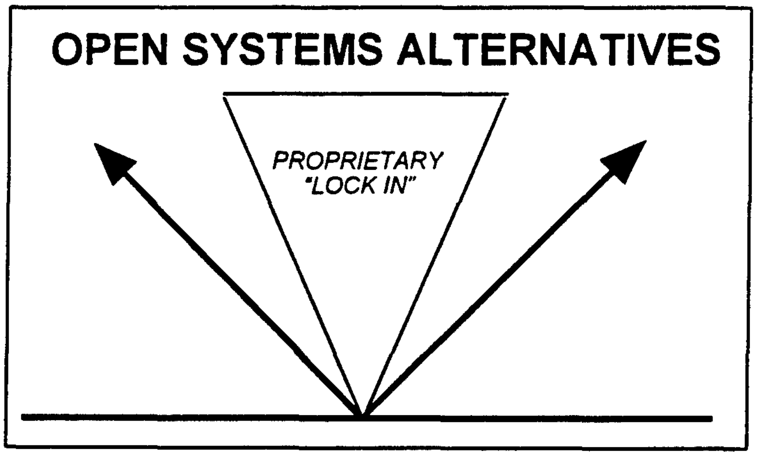


Figure 6-7. Standards: Degrees of Freedom

Overall benefit classification

Finally, an opportunity may be classified in terms of its overall benefits. These include the degree to which the opportunity provides possibilities for cost reduction and various categories of improved IT effectiveness. The following diagram describes this matrix classification.

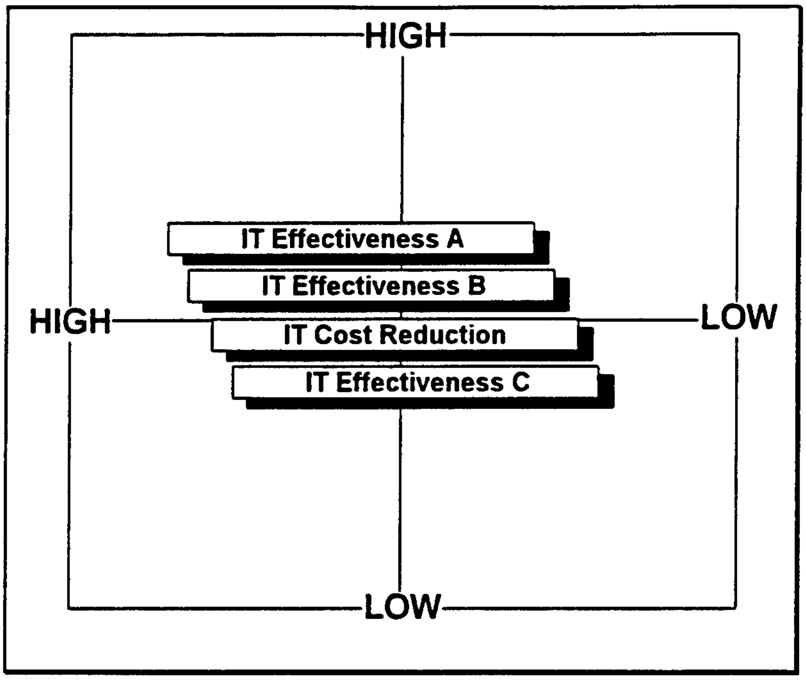


Figure 6-8. Benefit Matrix

The business case cost/benefit analysis process

Once opportunities have been categorized and classified, the business case cost/benefit analysis may be conducted.

Appendix F describes how the business case and the cost/benefit analysis could be constructed. A sample business case is provided, as well as the steps involved in building the case. These should be taken as only one way to perform this task. If the enterprise has other preferred approaches to developing cost/benefit analyses, they can be substituted.

Once the task is initiated, the AWG must review the baseline and target architecture documents developed in previous phases.

Migration planning

Upon review, the team selects a component(s) of the target architecture to consider for implementation and creates the action plans to implement that selected piece. In doing so, the work group must be careful not to lose track of the installed base of applications and technology. Few organizations can afford to scrap this investment and embrace open systems in a "flash-cut" fashion.

Instead, migration from old to new must be a gradual process. As the samples provided in Figures 6-3 and 6-4 suggest, these timeline issues must be considered as the team prioritizes its migration plans. Often, one project must be completed before another can begin. For instance, an organization may want to determine its DBMS technology before it identifies design generators or CASE tools. More examples of migration paths are included in Appendix E. While not a complete view of all types of projects that will be included in the migration options, Figure 6-9 depicts potential plateaus to migrate from an existing environment to one characterized by technology standards in the target environment.

Plateau costs

To assist in planning for the implementation of an IT architecture, it is useful to have a feel for the size of the effort in terms of staffing and costs. Unfortunately, at the architecture level, it is not possible to derive these estimates with a high degree of accuracy. It is possible, however, to apply past experiences in the form of "rules of thumb" and standard application development estimates.

Costs will crop up in a number of areas as a result of a series of projects. However, to arrive at a reasonable order of magnitude cost, we will focus on the following areas:

- The incremental computer processing and network hardware and system software needed to support the projects that will move the organization to the desired target architecture
- The application development and/or package procurement/modifications required to move to the target architecture
- The non-application initiatives needed to move to the target architecture.

Figure 6-10 is a sample summary of these cost projections by plateau and type of project as derived from the USMC SBA development project.

These ballpark estimates are intended to help strategic decision makers understand the resources required to properly evolve into the next generation of computing and reap all of the benefits that a strong IT environment brings.

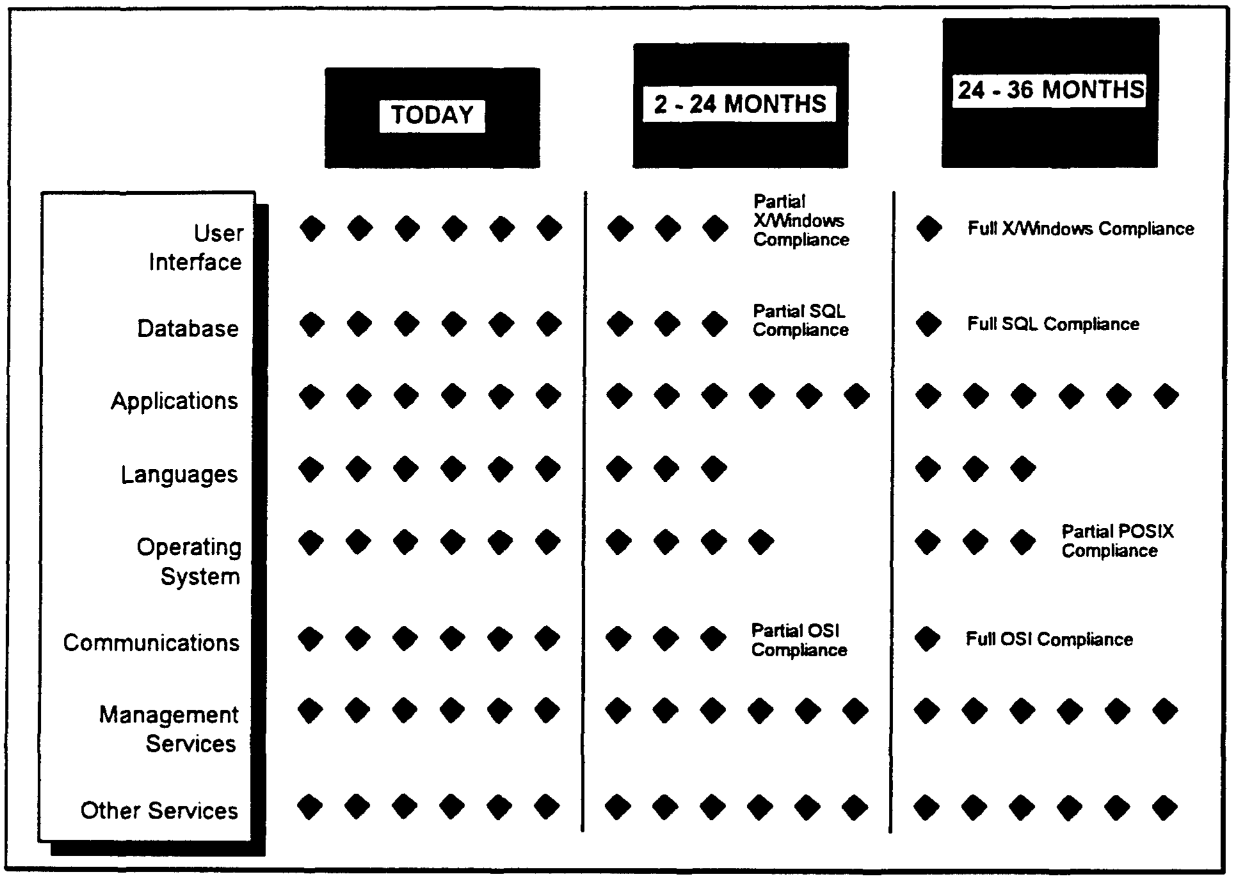


Figure 6-9. Standards Migration

Project Classification	Plateau 1 Estimated Cost	Plateau 2 Estimated Cost	Plateau 3 Estimated Cost	Total Estimated Cost
Application Development/Procuremen	$9M	$7M	$9M	$25M
Non-Application Initiatives	$2M	$0M	$0M	$2M
Computing and Network Facilities	$28M	$21M	$21M	$70M
Totals	$39M	$28M	$30M	$97M

Figure 6-10. Summary Ballpark Cost Estimates by Plateau

Creating and publishing the deliverable

The key deliverable in this phase is the *Migration Options Document* along with the high-level cost estimates. The migration plan will probably consist of three separate plateaus. Because of the unique time horizons in the Federal Government, it may be desirable to link the plateaus with the 2-year POM process.

Migration Options Document

It should focus on providing the ASC with a high-level understanding of the opportunities at hand while also providing business case backup information that justifies the proposed implementation opportunities and schedules. This document should also focus on highly visible short-term "payoff" projects to demonstrate the utility of this process along the way to the target.

After finalization and approval, the document is then delivered to the rest of the organization. The options document is extremely valuable to stakeholders who must prepare for the challenges that SBA implementation brings.

Effectiveness measures

- Organization's ability to accept and execute migration plans
- Rework required of the *Migration Options Document*
- Management's general acceptance of the plans.

In order to achieve management acceptance, the *Migration Options Document* must describe the basic elements of the undertaking (i.e., the major program components and initiatives).

The components should be such that they are easy to read and understand by functional area managers as well as upper management. They should not dwell excessively on the technical dimensions of the architecture, elements that should be included in a detailed implementation plan. For example, if a communications project is undertaken as part of the larger project, it would be appropriate to state that all buildings would be wired with token ring or Ethernet wiring, but it would be inappropriate to go into the details surrounding wiring closet issues and the link, or the time and dates they will be installed and which project team members would accomplish the task.

Technology and tools required

- Workstation and connectivity technology
- Word processing and graphics capabilities
- Dedicated workspace with clerical support.

Staffing skills required

- Migration planning expertise
- Writing and presentation skills

- Project planning skills and experience in assigning larger efforts into implementable "chunks."

Completion criteria

- Creation of high-level plans for each component of the target architecture
- *Migration Options Document* deliverable
- Management sign-off.

One of the hallmarks of information technology is that it constantly changes. IT managers are always confronted with one of two phenomena: The technology they have installed is made obsolete very quickly, or the technology they had forecasted never materializes. For this reason, we recommend that the *Migration Options Document* contain a contingency section to address these two dilemmas.

In essence, we recommend that each major architecture project contain an assessment of the technology and standard directions possible in the future. With that forecast, we recommend that users develop alternative scenarios for implementation should the technology or standards set forecasted for project implementation never materialize. We refer to this part of the process as "innovation-proofing. "

In the DoD, the other volumes of the TAFIM series—which deal explicitly with technologies, standards, styles of computing, etc.—are already in place and should evolve over time to provide a large measure of this innovation-proofing input.

No person or organization is entirely successful at predicting the future, but successful organizations will do it well most of the time by dedicating resources to technology forecasting and SBA administration.

Issues

- Consulting support needed
- Executive "buy-in"
- Workload of work team(s)
- Inventory scope management.

When plotting standards, there are other concerns to be addressed in the architecture. For instance, users may not want to "turn on" the proprietary extensions to open system products, such as relational database packages, because that single action moves them away from being open. While attractive functionality may be sacrificed, a passport to openness has been maintained. The team must keep this in mind as it consider its migration options.

- Consulting required
- Training required
- Key knowledge worker availability
- Existence and maturity of "open" technologies and standards.

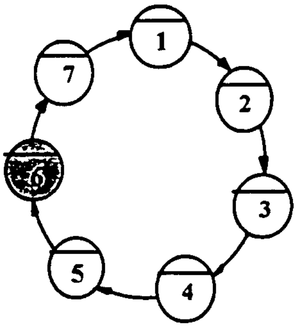
In many instances, one might find architectures based on evolving but currently incomplete standards. This requires that "workaround" strategies be developed. If the AWG regards standards on a continuum as we have recommended, this will not be as large a problem as it would appear at first inspection.

Section Seven: Implementation Planning

Contents **Page**

Figures

Section description



This section describes the overall process by which the AWG identifies and develops specific implementation plans for moving to the new target architecture. Included in this activity are descriptions of how the *Implementation Plan Document* is developed:

- Implementation project plans for Plateau 1 are developed.
- "Quick hits" for fast payoff projects are identified and pursued.
- Organizational communication mechanisms for promoting success are put in place as part of the architecture project effort and in anticipation of the SBA administration phase.

Objectives

To develop additional planning detail for the project initiatives identified as Plateau 1 of the *Migration Options Document*

- To define projects that can be completed quickly
- To create effective communication mechanisms for promoting success.

With the completion of the *Migration Options Document*, the SBA project is nearly complete. This section of the SBA Guide contains the process for developing the implementation plans for all Plateau 1 efforts.

The *Migration Options Document*, along with the other deliverables from prior phases of the SBA project, should be used to guide a detailed project scheduling process for the Plateau 1 initiatives, including specific delivery time frames and clear assignment of roles and responsibilities for each project.

At this point, the enterprise is well positioned to begin its transition towards the target IT architecture defined earlier in the SBA process. Enterprise project managers will be able to use these project plans as guides to development. The plans contain information about such issues as what is to be included in the project, the type of talent needed for the implementation team, and the infrastructure issues that may impact the success of the effort.

The plans do not define the total amount of resources required nor the project schedule because that level of detail can only be defined when each project is sanctioned by senior management. However, the details needed to get a project successfully off and running are certainly available within the plans.

As described in the previous section of this SBA Guide, the implementation projects have, by now, been organized into plateaus. Each plateau contains a set of interrelated projects in priority order. The plans that follow are for those to be tackled in Plateau 1.

Also included in this document are the project plans for a set of quick hits that the enterprise should strongly consider completing within the first year of its SBA implementation effort. The quick hit projects offer a good deal of benefit in a relatively short delivery time as well as providing a foundation for other Plateau 1 projects.

Individual project initiatives should then be kicked off with a preliminary analysis phase. In the initial design phase, more detailed deliverables will be developed showing a refined view of the information and system functionality through conceptual models and supporting documentation. Also, a refined cost and benefit estimate should be made at this time for each project allowing a "go/no-go" decision to be made on a project-by-project basis, considering all of the interrelationships defined in the architecture deliverables.

This phase is based upon the very simple notion that if an architecture does not begin to deliver concrete benefits in under 12 months, it has a low probability of being implemented overall. As a test of its real world viability in today's world of results-oriented management and reward, a program must be able to deliver a concrete payoff project to ensure that a manager's year-end personal objectives are met or the program will not be implemented. For this reason, it is key that short-term payoffs are identified and implemented early on in the architecture process. Once these "small wins" have been put into place, this phase focuses on broadening awareness throughout the organization to induce "culture change." Mid-term benefits are then harvested and a benefits measurement program is put in place for the duration of the program.

Scope

To define the plans necessary for migration, with an emphasis on quick hits, while the longer-term strategic standards-based architecture is developed and implemented. The document has a short-term payoff orientation.

It is recommended that, if the AWG wants to deliver a detailed technical implementation plan, technical and operational professionals be introduced as key players during this phase.

A natural question arises from this approach: To what degree should the AWG be involved in detailed project management? The answer depends upon the size and scope of the implementation project. It is recommended that the "Level 1" high-level project plan be developed by the AWG, and that more detailed project implementation plans be managed within the operational or business units in which they logically reside. Progress updates may then be delivered to the AWG and ASC. Figure 7-1 illustrates this relationship.

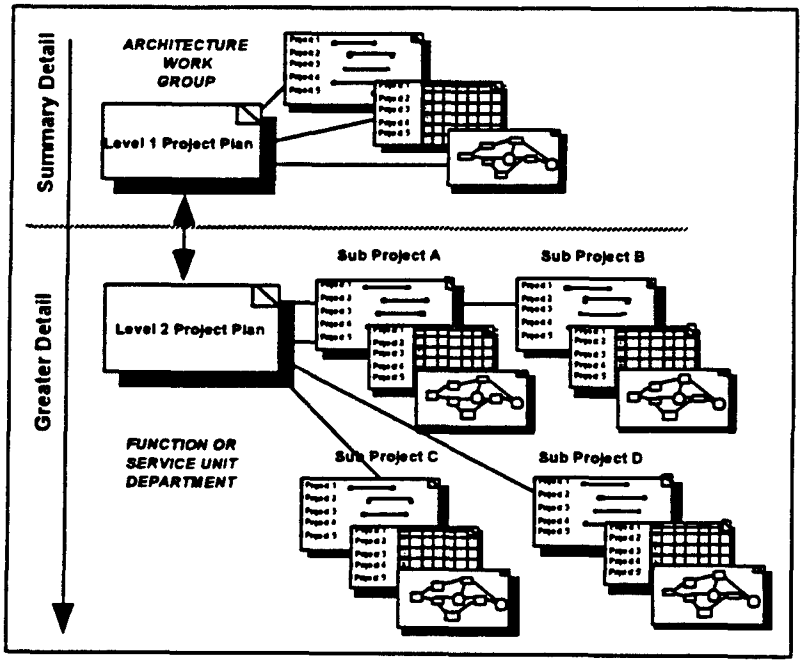


Figure 7-1. Levels of Implementation Planning

Most organizations have key technical leaders on the AWG, and detailed implementation plans are most successfully developed within the discrete operational units in which they are being implemented.

Some AWGs may elect to split implementation planning into two levels of activity: a high-level architecture implementation plan and a secondary technical implementation plan.

Deliverables

Implementation project plan documents that contain the detailed road map for migration to the *Implementation Plan Document*. A sample outline for this document is included in Appendix I.

During the migration options phase, a series of migration steps were outlined. In this phase, the team characterizes the size and scope of implementation plans and the timing of the projects, as well as developing alternative contingency plans.

Critical success factors

- Project management and estimating skills
- Detailed planning talent on the team
- Team that is comfortable in working with a short-term focus.

Standard implementation planning techniques that should be used during this phase have not been discussed. It is assumed that the reader will be familiar with these techniques in the same manner in which he/she understands other processes such as data modeling (which is likewise outside the scope of this document, although examples are provided).

Throughout this document, the focus has been on the need to identify opportunities that provide concrete payoffs in implementation. If an architecture does not provide initial payoffs, there is a high probability that the entire architecture will never "see the light of day." The following needs have been described:

- A short-term focus combined with a "fast path process"
- An architecture and attendant implementation based on discontinuous, chaotic business realities of today's "fast cycle" organization

- Implementation projects that provide project-oriented deliverable payoffs rather than "grand strategic" payoffs some time in the distant future
- An ongoing process that defines architecture and standards with room for entrepreneurial improvisation and implementation.

SBA implementation must possess all of these qualities. For this reason, it is recommended that, in addition to standard project planning techniques, the AWG focus on several other aspects of implementation to ensure successful implementation.

"Quick hits": Implementation of short-term payoffs

There is more than a grain of truth in the saying *"in the long run, we're all dead."* Nowhere is this more true than in implementation planning. In today's typical organizational culture, short-term (3 to 6 months) payoffs are required as a condition of employment and advancement. If the entire implementation program is to be a success, it must contain a minimum of one major implementation activity that is an integral part of the SBA plan and may be capable of being implemented in a short time frame. It must be of sufficient significance that its implementation will assure the AWG members (or their management) of attaining their annual program goals and objectives.

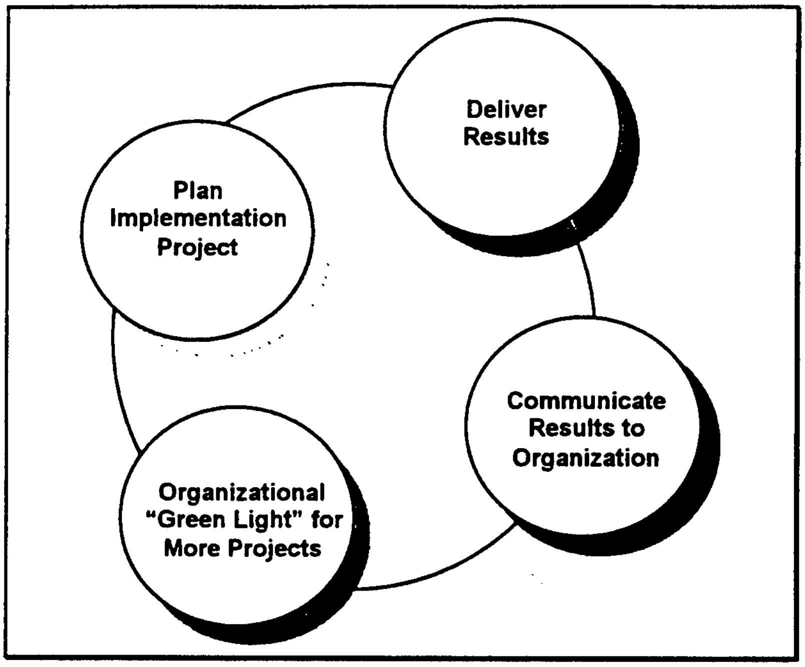
When implementation activities are linked to the enterprise's reward system, things get done and heretofore non-cooperative organizational task force members begin to make things happen.

The other central objective of providing a short-term payoff is that the successful implementation may then be used as a pilot case example for the rest of the organization of how a standards-based architecture can provide immediate benefits, and that truly major benefits will accrue to the program if it is followed over time.

Communication: Organizational awareness programs

Upon identification and implementation of a major short-term payoff opportunity, the AWG should spend a significant amount of time conducting a "public relations advertising program." Figure 7-2 illustrates the recommended process that AWGs should follow to ensure that the organization is behind the implementation effort throughout the SBA life cycle.

It is recommended that workshops or presentations be continuously conducted throughout the organization after the AWG has a solid implementation success on its hands. People or processes that actually "get something done" are rare in most organizations. If projects are successful, there is a great likelihood that the architecture planning documents will be read and implemented throughout the organization.



Deliver Results
Plan Implementation Project
Communicate Results to Organization
Organizational "Green Light" for More Projects

Figure 7-2. The "Results Communication" Cycle

Architecture plan modifications

The AWG's designated implementation team will make ongoing modifications to the overall process as it progresses in implementation over time. There are times when individual implementation projects blow up or need to be terminated. Sometimes these projects are outright failures due to poor management or resource constraints and the like. At such times, it is sometimes convenient for management to conclude that the "architecture is fundamentally flawed." Thus, important projects are sometimes eliminated because of subproject deliverable failures. The overall architecture becomes, as it were, the fall guy for a poorly implemented project.

It is recommended that such failures be carefully evaluated in the context of the overall architecture project implementation cycle before changes are made to the overall architecture. In nine cases out of ten, implementation strategies and tactics will require adjustment, rather than the overall architecture. However, sometimes failed projects do show opportunities to improve the overall architecture.

Because the architecture is developed on a group consensus basis, making significant changes requires ASC sign-off. In theory, one aspect that will not change is the architecture principles. *These provide the "constitutional" backdrop to the overall standards-based architecture.* If the organization does discover that some principles must be changed, then the equivalent of a "constitutional amendment" process must be developed by the AWG and approved by the ASC. Figure 7-3 illustrates this process.

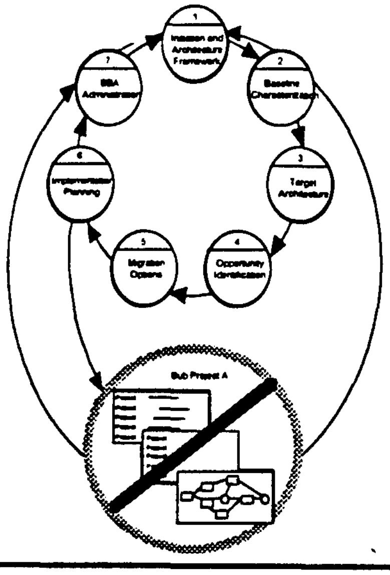


Figure 7-3 Project Impact on the Architecture

Constraints

An inexperienced implementation planning background will limit the team's ability to develop effective plans.

The degree to which highly granular implementation plans are developed will depend upon the skill set and experience of the AWG. If the AWG is more highly skilled at planning versus implementation, it might be logical to identify business or service unit department-level personnel to actually carry out the detailed implementation planning discussed in this phase.

Task list

- Initiate task
- Assign team to build detailed implementation plans for Plateau 1 projects
- Develop cost/benefit case by project
- Produce implementation plans by project
- Develop security implementation plans by project as necessary
- Identify standards implementation strategy by project
- Identify key interrelationships and dependencies among projects
- Establish timeline for each project
- Draft *Implementation Plan Document*
- Conduct review with ASC
- Finalize *Implementation Plan Document*
- Distribute *Implementation Plan Document.*

Creating and publishing the deliverable

Implementation Plan Document

The key deliverable out of this phase is the *Implementation Plan Document.* It should focus on providing the ASC with a detailed understanding of the projects being developed as well as all traditional project management reporting techniques. It should include:

- Major project descriptions
- Milestones and project interrelationships
- Resource requirement definitions
- Project deliverable definitions
- Key responsibilities and accountabilities by project and program.

Effectiveness measures

This phase will vary widely in terms of calendar time required for completion based on project size, scope, organizational culture, individual schedules, and the resources required to perform the project. We recommend that the implementation project teams constantly remind management of the need for a "fast path" implementation to ensure rapid deployment of project implementation efforts. Effectiveness measures include:

- The ability of the plan to show continuous improvement and results
- The degree to which implementation plans can be developed
- Management enthusiasm regarding opportunities identified
- Timeliness of project implementation.

The ASC should be kept informed of all status activities as mentioned previously in this section. It is this group that will keep pressure on their management groups to ensure that projects are implemented successfully.

Tools required

- Word processing and graphic presentation packages
- Project planning software tools
- Spreadsheet tools and/or user-friendly personal computer-based database packages for inventory logging.

The key deliverables out of this phase are the individual implementation plans themselves. Therefore, project planning tools, as well as those described above, will be required for the task at hand.

Staffing skills required

- Migration planning skills
- Project management skills
- Writing and presentation skills
- Familiarity with word processing, presentation, spreadsheet, and database packages that run on most popular personal computers.

This phase requires individuals who are well-seasoned individuals in the art and science of migration planning and project management. If the AWG does not have members with these traditional skills, the team may be augmented on a temporary basis with personnel outside the team.

Completion criteria

- The development of all short-term and mid-term project plans
- Management review and acceptance.

Successful, on-time implementation of projects identified during the implementation planning effort is the sole measure of how well the completion criteria have been met.

In addition, the degree to which middle- and long-term opportunity projects are pursued is key to the successful implementation of the overall architecture. Frequently, such initiatives get dropped before "the war is won." With the focus on short-term payoffs, it is critical that the ASC not abandon its efforts after early "successes."

Issues

- Project management skill capabilities
- Workload of work team(s)
- Business case criteria acceptability
- Consulting required
- Training required
- Subject matter expert availability.

Resource constraints may make project implementation a challenge for both the ASC and the AWG. It is very important that all of the issues outlined on the list above be addressed in reviewing all implementation plans.

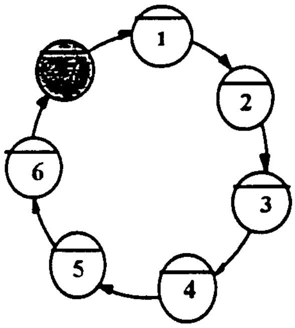
This page intentionally left blank.

Section Eight: SBA Administration

Contents **Page**

Figures

Section description



In the SBA administration phase, the process by which the organization maintains its new IT architecture is identified. The SBA administration process defines the procedures, human resources, and communication devices needed to keep the plan current with the organization's mission and priorities. Because this process is an integral part of the IT planning effort, it is essential that personnel be dedicated full-time to architecture administration.

This section describes the overall process by which the AWG monitors and checks the success of the new target architecture. This is a key activity, as the team seeks to continuously improve the development and implementation of the IT architecture. Included in this activity are descriptions of the need for an ongoing architecture administration process and of how an *SBA Assessment Document* is developed:

- An SBA management team (SBAMT) is recommended to maintain the SBA.
- An SBA development project review process is developed.
- An ongoing process is developed for the measurement and monitoring of project problems and architecture compliance.
- An ongoing process is developed for keeping the SBA document alive.

The output of ongoing architecture reviews is a self-critical document that is used to modify the architecture documents produced in prior phases to "keep them alive." As such, this phase is the last in a continuous cyclical improvement process. As Figure 8-1 suggests, it provides the organization with a way to learn from past mistakes and make adjustments to future plans to ensure its ongoing success.

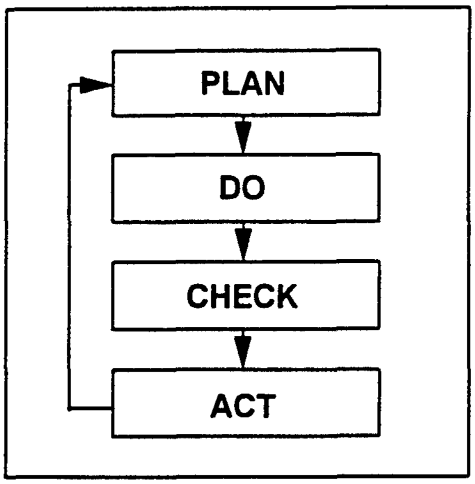


Figure 8-1. The Continuous Process Improvement Cycle

Objectives

- To create a measurement process for the remainder of the SBA implementation
- To review the results of project implementation
- To modify current plans based on actual experience
- To integrate the SBA process into the mainstream planning and management activity.

Now, more than ever, it is important that IT professionals have plans that work when implemented. The plain fact is that some plans *do not work*. A number of contributing factors result in the half-implementation or failure of architecture plans. The most common one is that management changes direction, and the attendant technology priorities change as well. Other times, plans are not implemented because of flaws in the planning process itself—some of which have been discussed in this SBA Guide. Architectures are frequently not implemented because either the recommended technology does not deliver the solution or the technology *"never shows up"* (also known as technology lag). In the latter case, a vendor's technology promises never materialize in the marketplace. This happens with both technology and standards themselves.

The final phase in the SBA planning process is to "reality-check" the architecture to ensure that the original design criteria are bearing results. The best way to accomplish this on an ongoing basis is to fully integrate the SBA planning process into the mainstream management practices within the enterprise.

Scope

- All projects defined in the implementation plan are within scope of the evaluation.

This step is executed once the implementation plans for Plateau 1 projects have been approved. This is an optimal time for effecting the transition of the SBA process from the "experimental" arena into the mainstream management function. By establishing a credible position within the management function, the projects coming out of the SBA process will have greater likelihood of funding and implementation. It may be a significant challenge to become a full-fledged component of the general business planning process, but anything short of this status is associated with risks to the projects and to the SBA process itself. The ability of the SBA process to achieve such status will, in many cases, reflect the success of the initiation phase that launched the SBA in the first place.

After a reasonable period has elapsed in the implementation process (or, alternatively, as a direct follow-on to the delivery of approved plans), the AWG should conduct a brief review of the projects defined in the *Implementation Plan Document* to ensure that those projects' objectives are being met and that payoffs are being obtained through the implementation process (see Figure 8-2.). We refer to this as a "process check" and, as such, it will provide a *quality assurance* dimension to the overall planning process.

This process check of the architecture should occur on a cyclical basis throughout the IT planning process. This check should focus on the deliverables of the architecture, as well as on the architecture process itself, and be modified accordingly.

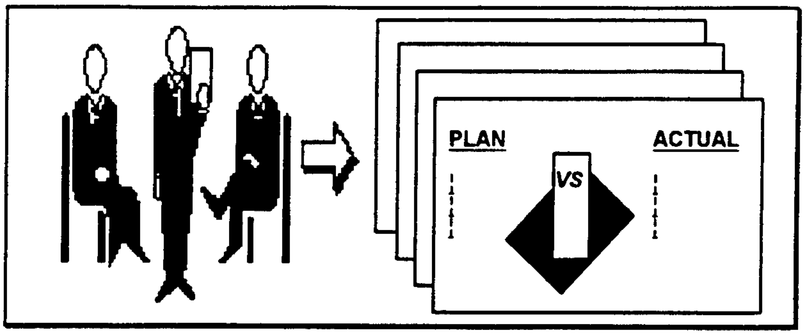


Figure 8-2. The Team Reviews Each SBA Project Plan

Deliverables

- Establishment of the SBAMT and recommended processes to keep the SBA "alive"

SBA Assessment Document (at a later date in the implementation cycle, but after the SBA development project concludes)

The architecture is subject to regular assessment and an *SBA Assessment Document* is produced at each review. The *SBA Assessment Document* may be developed by the SBAMT. The process of SBA implementation, however, is ongoing and subject to the organization's commitment to continuous process improvement. (Quarterly reviews are recommended in the first year, semi-annual reviews thereafter.) A sample outline for this document is included in Appendix I.

Critical success factors

- The organization must be willing to sponsor the SBA process as an ongoing management activity.
- A team review of the process that solicits organizational buy-in must be used.
- Time must be dedicated to this effort.
- Key knowledge workers must participate as required.
- Results must be communicated.

Modifications to existing plans must be made.

It is essential that the organization establish a review process and dedicate resources to the effort. Perhaps the greatest reason for implementation failure is the simple, but often overlooked, requirement to obtain organizational buy-in and make the architecture implementation process a team-based effort.

Much of this activity is organizational and political. In the end, politics is the art of inclusion. Any success enjoyed early on in the initiation phase will contribute to continuity of the process now that the process deliverables have all been approved. The real challenge starts now as implementation plans are to be put in place. If initiation was not successful, there remains much to do in positioning the SBA process within the organization. Any organization pursuing standards on a managerial dictatorship model will run a much higher probability of failed implementations than the more team-oriented process that has been outlined throughout this SBA Guide.

In the area of standards-based architecture, it is paramount that the AWG build into the overall process a review system to ensure compliance with the objectives set out by the *Architecture Framework Document, Baseline Characterization Document, Target Architecture Document, Opportunity Identification Document, Migration Options Document,* and *Implementation Plan Document.*

Constraints

- Fear of being labeled a failure can undermine this effort.
- Other priorities can also limit the effort that participants can dedicate to this project.

As Figure 8-3 suggests, the key to success in establishing an assessment process is to have the plans owned by as wide a team as possible across the enterprise (rather than a set of individuals with "agendas" ready to assign blame for failure).

Perhaps the largest constraint is management's unwillingness to dedicate the resources needed to keep the SBA in the forefront of activities in the systems development and/or work redesign arena. The ASC must be ready to address this issue in order to create the team-oriented environment necessary to make SBA a success.

In this new kind of environment, assessment and review become less personally and politically charged. The result is that the assessment process becomes easier to successfully conduct. This form of organizational behavior also encourages successful implementation in the first place.

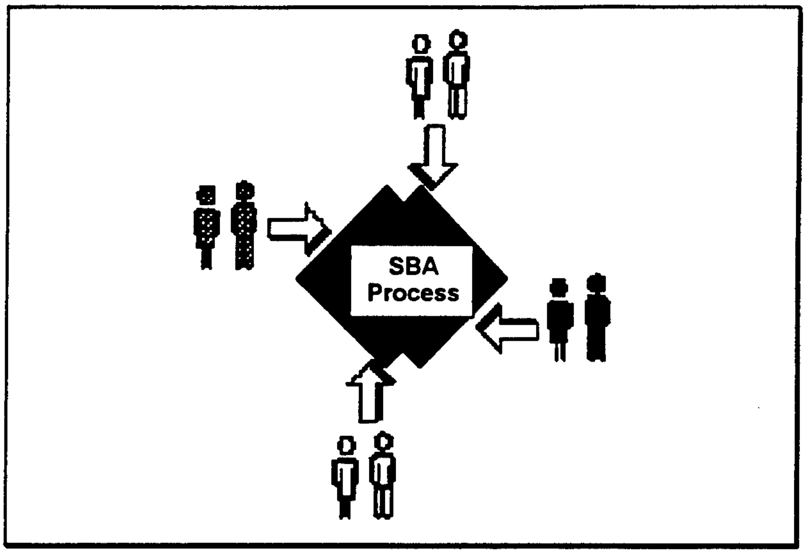


Figure 8-3. The Entire Organization Should Be Included in the SBA Process

Task list

- Launch the implementation plan, staff the SBAMT, and establish the ongoing process to be followed.

The following tasks can only be done after some progress has been made on the project initiatives defined in the SBA implementation plans (i.e., after the SBA development "project" has concluded).

- The SBAMT maps results against the *Architecture Framework Document*, the *Implementation Plan Document*, and their measurement criteria.
- Current project and future plans are reviewed.
- Appropriate modifications are made and distributed to the review committee and the appropriate project managers.
- "Lessons Learned" are developed and included in the *SBA Assessment Document* for distribution.

The first step in the assessment process is to establish an SBAMT. This team should be staffed with experienced planners and technologists who have a deep-rooted understanding of the implementation projects.

Once established, the team must conduct a general assessment of the projects to see if, in fact, the projects are being implemented. This is done by mapping the results against the implementation plans and asking some hard questions, such as:

- Is the architecture framework still valid? Should any of the architecture principles be modified? Which ones and why? What has changed?
- What were the benefits of the identified projects? Cost savings, value-added benefits, or softer long-term intangible benefits?
- Have adopted standards been materially implemented in the organization? How far along has the standards road been traveled? How far, given this process check, do we have yet to go? Have we gleaned 80 percent of the benefit already or is there still significant payoff down the road?
- Does the organization recognize the payoff that has been achieved?
- Given the current state of implementation, have any other payoffs been obtained that may not have been originally predicted (the *Opportunity Identification Document* should be reviewed in this context)?
- In general, do the plan's standards appear to be changing?
- Have any standards, targeted as important, not yet matured as much as was originally anticipated by this point in time?
- What is the status of the technology that was selected for implementation? Has it "shown up on time" in the marketplace?

After these questions have been answered, adjustments to the original plans should be made (i.e., if implementation is not working for tactical reasons, specific steps will have to be developed to produce "workarounds") and reviewed with the ASC.

After review of the plans, the team should step back from the assessment and begin to analyze the exact cause of the shifts of emphasis. These "lessons learned," together with the modified plans, become the *SBA Assessment Document* (see Figure 8-4.).

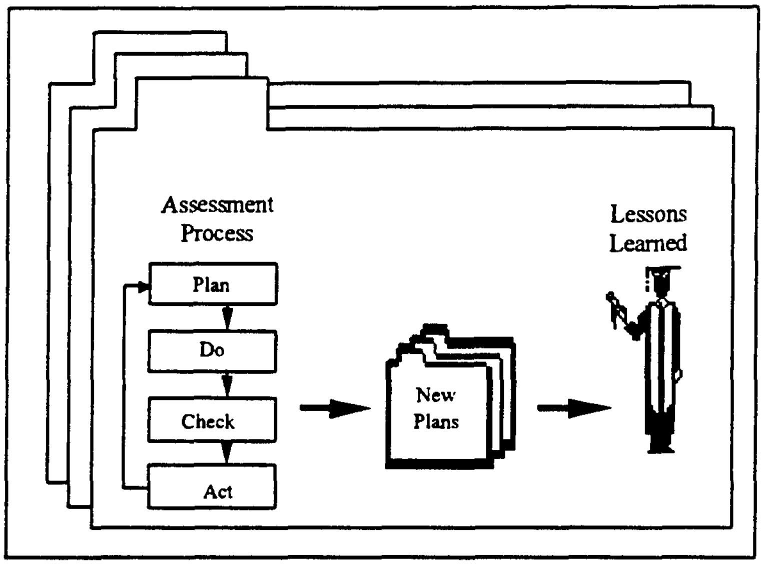


Figure 8-4. The *SBA Assessment Document* Includes New Plans, Revisions to Old Plans, and Lessons Learned

Often overlooked, documenting the lessons learned becomes very valuable to the review team when defining the modifications to future plans, and it helps future implementation teams to "not make the same mistake twice."

Effectiveness measures

- Organizational buy-in to the process measured by active and enthusiastic involvement
- Implemented architecture attributes are measured and assessed
- Ease of plan modification
- Communication mechanism.

The assessment effort can be judged by the degree to which the assessment team can examine SBA results to date and determine the appropriate actions to take to keep the architecture process on track. Ultimately, the effectiveness of a given assessment can only be measured by the success of future implementations.

Technology and tools required

- Workstation and connectivity technology
- Word processing and graphics capabilities
- Dedicated workspace with clerical support.

Staffing skills required

The assessment team is typically composed of members from the original SBA AWG and a few implementation project managers. While it is true that this team meets only a few times a year, management must be prepared to reassign workloads and the like because sometimes the effort needed to complete the assessment can be quite extensive. The staffing skills required include:

- General knowledge of SBA
- Planning skills
- Subject matter experts (as needed).

Occasionally, the assessment team will need to rework existing plans. They will need to call upon key knowledge workers who have working knowledge of specific projects or technologies. Management must be willing to commit what it takes to keep the SBA process alive and on track.

Completion criteria

While architecture assessment is an ongoing effort, a particular review cycle can be considered complete when:

- Each SBA project status and plan has been reviewed and compared against the architecture principles and target architecture plans.
- Lessons learned have been documented.
- The completed assessment document has been reviewed and approved by the ASC.

Ultimately, it is the ASC's decision as to when a given assessment effort is complete (i.e., the committee is responsible for the success of the SBA effort as a whole).

Issues

- Training needed
- Consulting needed
- Remodeling the core architecture may become necessary
- Time must be spent changing the culture such that reviews are seen as a process improvement vehicle and not as an exercise in "pointing the finger."

Besides the training and consulting support needed for proper architecture assessment, there are two major issues that can impact this phase: The extent of *architecture remodeling* needed and the management of cultural *change*.

Architecture remodeling

When should you remodel? When one of the architecture principles has changed. Another reason for remodeling could be that a major change in technology took place that was so significant that your architecture plans did not anticipate it; however, this will become increasingly rare. One of the major benefits of standards planning is that standards, unlike the underlying technology itself, change far less frequently.

In theory, you should never have to change your architecture framework if the architecture principles never change; however, they do change from time to time. When this happens, the review team should discuss and confirm the perceived changes with the ASC.

If necessary, the committee can sanction a task force to do necessary rework of the affected SBA documents. However, this step is usually not required, because the assessment team is more than likely composed of the same personnel who developed the original plans.

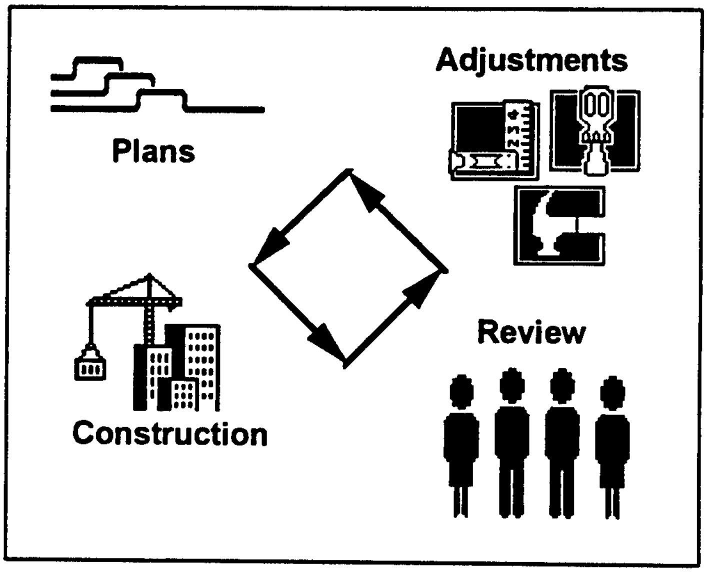


Figure 8-5. The Organization Must Gain a Working Understanding of SBA and Learn to Appreciate Its Value

Cultural change

As a final note, the review process described in this section should not be taken lightly. It is central to building and maintaining a solid SBA. Because of its importance, the DoD community should dedicate resources to the promotion of, and education in, standards-based architecture.

The goals of the promotion and training program should be to expose the entire organization to the change process and familiarize personnel with the benefits inherent to open systems. In so doing, the "gut-level" values of the organization will change and SBA management will become everyone's business.

Appendix H contains a more detailed example of the kinds of processes which may be recommended for SBA administration at the end of the SBA development project.

Appendix A: How To Do Architecture Principles

Foundation of a standards-based architecture

Architecture principles are statements of preferred architecture direction or practice. They are simple, direct statements of how an organization wants to use information technology in the long term for 5 to 10 years. They establish a context for architecture design decisions across an organization and help translate business criteria into a language that technology managers can understand. Each principle is accompanied by a statement of the rationale for the principle and a statement of the principle's implications.

Many organizations skip the principles definition process and jump right to modeling their architectures and setting standards. The result has often been a technical myopia–organization focus on technology selection issues and never deals with how they are going to manage the technology until a selection of an unpopular vendor or technology raises the issue to a head.

The "IT constitution"

Principles allow for diverse business, operational, and technology personnel in the enterprise or work group to develop a common language and shared understanding of the challenges facing the organization. Architecture principles become the "constitution" by which the overall architecture is designed and implemented. In theory, principles change unless, like the U.S. Constitution, they are amended through a formal amendment process. This process was described in the previous sections.

Architecture principles are the foundation of a standards-based architecture and are necessary to achieve the degree of organizational consensus and understanding required to move ahead with an integrated, standards-based architecture. Experience with architecture principles has shown that a more open, standards-based environment is often the result of a principles definition process. Principles also provide organizations with a stable base from which to make decisions. Principles change as the organization's mission or business changes—often relatively slowly. They provide a framework against which to test later decisions and guide subsequent procurement and implementation decisions.

For some time now, many leading practitioners and academics have been arguing for a generic approach to principles. Principles can be especially powerful in helping an organization move to a new technology architecture; for example, the benefits achievable through a network computing environment enable the adoption of new classes of principles. Additionally, an appreciation of the case for standards-based architectures enables the "driving down" of principles to standards and guidelines, which can enable the actual implementation of systems. Consequently, the reader will note that the following discussion of principles has a unique thrust.

Establishing a coherent set of architecture principles is therefore critical to forging a standards-based architecture. Principles force enterprises away from individual discussions of vendor products to focus on the desired behavior of the architecture. Principles provide a vehicle for key stockholders to discuss and agree upon how they will organize and implement information technology.

A principle may deal with any aspect of architecture; for example, a principle that deals with information architecture may be:

"Business terms and associated data element definitions should be defined consistently and be readily available to users throughout the organization."

A technology principle might be:

"All computing and communicating devices should interconnect through a common networking environment that is based on industry standards. It should support interconnection among internal units and with users, suppliers, and other business partners."

The definition of principles can be influenced by a number of factors: current policies, business drivers, strategic business decisions, IT trends, existing architectures, and organizational practices. We have found that principles generally fall into five categories:

- Principles that affect all aspects of IT (meat-principles)
- Work organization
- Information
- Applications
- Technology.

Although principles are the foundation of an architecture, they are not a complete architecture as illustrated below. A thorough analysis of how technology will be deployed and what viable vendor products and industry standards are available must be performed before technology can be procured or systems can be delivered.

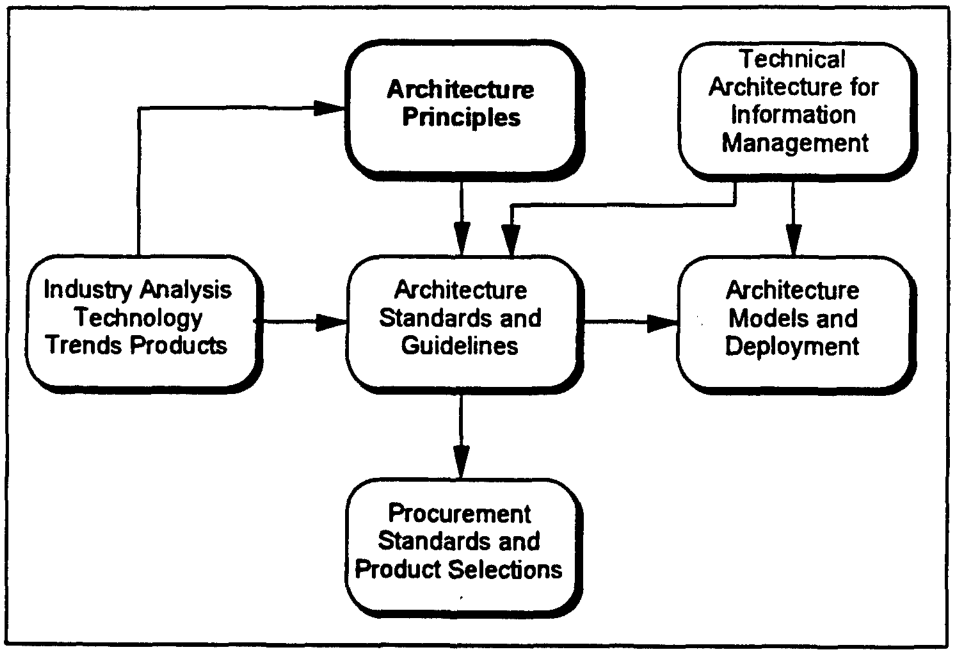

Architecture Principles
Technical Architecture for Information Management
Industry Analysis Technology Trends Products
Architecture Standards and Guidelines
Architecture Models and Deployment
Procurement Standards and Product Selections

Figure A-1. Relationship of Architecture Principles to Standards

The remainder of this appendix discusses how principles begin to define a style of computing, how principles are defined, and provides a generic list of principles that can be used as the basis for defining a customized set of principles for an organization. It is important that the organization develop its own principles and not simply duplicate those listed here, since the value of the exercise is the group consensus and discussion around these key issues.

Styles of computing

Principles are analogous to zoning laws. Zoning laws establish a set of rules for the usage of land (setbacks, building size, etc.) and for the type of building that will be put on the property. Like zoning laws, principles tend to change relatively infrequently. Likewise, architecture principles set rules for how IT will be used, guide implementation of systems, and begin to define a "style of computing" that an organization will undertake.

An organization's computing style has a number of dimensions:

- **Dispersion**–To what degree will control over IT be dispersed to business units and departments within the organization? How much autonomy do business units have about decisions on applications, data, and technology?
- **Distribution of applications and data**–Will applications and/or data be centralized or will they be placed close to the user?
- **Decentralization of technology**–Will the technology environment be mainframe-based? Will it be highly decentralized and integrated around a network? What is the role of intelligent workstations?
- **Proprietary or open.** Will the architecture be based on a vendor's product approach (e.g., AS)? Will it be based on industry standards? To what degree?

The principles should articulate the organization's view on each of the dimensions. If successfully articulated, the principles can simplify many subsequent modeling and standards decisions.

Principles and their relationship to open systems

Principles often promote a shift to a standards-based architecture. First, when organizations go through the principles definition process, they begin to articulate the valuable characteristics of their desired architecture. Characteristics such as reusability, common components, interchangeable parts, and increased modularity of the architecture are often stated in principles.

When discussing the implications of principles, many organizations begin to see open systems and industry standards as at least partial solutions. Articulating the architecture principles provides a way to discuss “openness” as a desired attribute without getting into a battle between the proprietary and open camps that exist in many organizations.

The process for creating principles

Creating principles is inherently a dynamic, consensus building process. One senior IT executive characterized it as “social engineering” by providing a forum for a diverse group of IT and business unit managers to gain consensus regarding what is to be done and how it will be done. Most organizations find that they can adequately articulate their architecture direction in thirty to forty well-thought-out principles.

Creating principles is a five-step process to be conducted within the first phase of the SBA planning process, architecture framework:

1. Establish a principles task force within the ASC

The first step is to create a task force within the architecture framework phase that includes a mix of both IT and business unit personnel that represents the organization as a whole. This group functions as a subcommittee of the overall ASC. Development, operations, data management, and planning functions from the IT community should be represented. Business and operational unit representatives should be chosen who can speak for operational units. If there are tactical considerations, such as boundary interface definitions and the like, they should be an integral part of the unit as well.

It is important to have decentralized (dispersed) IT and business units represented as well as operational and tactical constituencies. While this may result in a large task force, the value of getting broad buy-in to the result is critical. A good task force size in a large organization is about ten people; however, task forces as large as *sixty* people have successfully defined principles. The process must be kept moving. If it bogs down, the commitment of task force members will disappear.

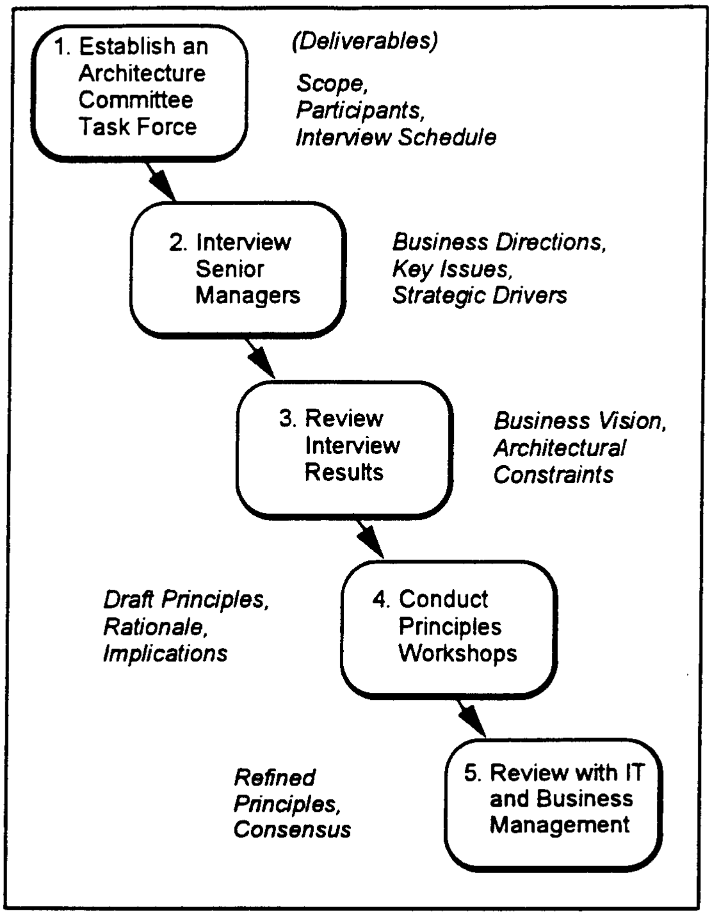

1. Establish an Architecture Committee Task Force
(Deliverables)
Scope, Participants, Interview Schedule
2. Interview Senior Managers
Business Directions, Key Issues, Strategic Drivers
3. Review Interview Results
Business Vision, Architectural Constraints
4. Conduct Principles Workshops
Draft Principles, Rationale, Implications
5. Review with IT and Business Management
Refined Principles, Consensus

Figure A-2. Architecture Principles Process and Deliverables

Once the task force participants are defined, the next step is to hold a workshop to introduce examples of architecture principles, the process that the task force will be going through, and how the task force will be organized. If the task force is large, it may be broken down into different topic areas such as the ones identified above (overall principles, IT organization, information management, application management, and technology management). The examples of principles discussed in this appendix can be used as "straw man" examples.

Next, the task force needs to identify the senior business managers to be interviewed. These are managers who can discuss the key business initiatives of the organization and the major directions that the organization will be taking in the next few years.

2. *Interview senior IT and business and operations managers*

Interviews are conducted with the business or operational unit managers. The objective is to understand the key business issues, directions, and constraints that the organization is dealing with and the organization's view of IT. For example, what is their view of the role of IT? Strategic or purely tactical support? How much risk are they willing to take with IT? Do they view IT as providing value, or as an additional cost of operating? How much control and autonomy do they want to exercise over IT decisions? What kind of time frames are they planning within (one year, five years, longer)?

3. *Review interview results with ASC*

The next step is to use the input from the interviews to identify the overall role of IT and to define the business and organizational constraints on IT. Information on the exist-ing IT environment is valuable here, as it may constrain the principles or make some principles unrealistic.

4. *Conduct principles workshops*

Once the task force understands the constraints and plans, it can begin to work on the principles themselves. The topic areas discussed throughout the rest of this appendix are a good starting point, but the principles need to be stated in the organization's own words, discussed, and agreed upon by the participants. Some characteristics of good principles are:

Principle Characteristics

1. They clearly state a fundamental belief of the organization.
2. No motherhood! Each principle should have a counterargument; for example, "information is an asset" is not a good principle, because it is hard to disagree with it.
3. They should be simply stated and understandable to both business and IT managers.
4. They need to have rationale. Why did this principle get stated this way? What alternatives were discussed?
5. The implications need to be discussed and documented; for example, what impact does this principle have on the IT organization? On management processes? On technology?
6. They conform to Federal mandates.

Figure A-3. Characteristics of Good Principles

It is also important to keep principles at the correct level. Too often organizations get into too much detail and actually end up defining standards and technology choices. That comes later, when input on the installed base and target architecture is available.

The following is an example of a principle, its rationale, and implications:

Principle

Our systems should utilize standard, shareable, reusable components across the enterprise.

Rationale

It is critical that the IT organization improve its response time to business needs and delivery systems faster and with better quality. Our organization is going through substantial change and IT must be better able to build flexibility into its systems and allow them to adapt to changing business requirements.

Using standard components as the basis for defining and building the architecture and delivered systems can improve our productivity by using previously defined and built components. Rather than build new components each time, developers can concentrate on new business requirements, rather than redoing existing work. We believe that the ability of our systems to adapt to changing requirements can be improved by using standard components.

Implications

There are a number of management and organizational implications from this principle:

- A means of coordinating, defining, and communicating the available standard components will need to be developed.
- Areas where definitions of standard components will be required include business processes, applications (at all levels), and technology components (processors, system software, network components, languages and development tools, and data, such as subject databases, conceptual designs, physical implementations, etc.).

A management process will be required to track the generation and usage of these shareable components and to standardize them where needed.

- A standard definition of each component type will also need to be defined. This could be facilitated through a well-implemented common system delivery methodology.
- A library of definitions, terms, access rules, characteristics, and interrelationships of each of the application, information, technology and, potentially, organizational and business components needs to be implemented corporate wide.

Figure A-4. Sample Principle

Meat-principles

Meat-principles are principles that apply to the IT environment as a whole. They address the organization's position on architecture, migration, and risk management, as well as its orientation to open or proprietary systems.

Architecture focus and compliance

Organizations have different views regarding how much they are willing to spend for an architecturally compliant environment. Some organizations believe that the potential additional cost of architectural compliance outweighs increased short-term costs. In other cases, cost pressures or a shorter-term view of benefits will reduce the impact of an "architected" environment.

Systems and technology infrastructure implemented by our organization will be compliant with our architecture even though there may be an additional cost for architectural compliance.

Implications

Agree ←	→ *Disagree*
• Faster migration to new infrastructure	• Slow migration—probably will not implement architecture
• Longer-term view of benefits	• Shorter-term view
• Lessened dependence upon existing installed base of hardware/software	• Probably stay with existing vendor/product set—more oriented toward existing proprietary systems

Cross-functionality

Several organizations have seen the opportunity to reuse applications, data, and related infrastructure in similar type functions across the organization. This requires a broader view of the business and an understanding of how to identify similar functions across the enterprise.

An orientation toward identifying and implementing cross-functional systems creates an opportunity for standards, standard components, and open systems. Technical integration opportunities are identified later on when developing architectures based on such principles. Portability of applications and data become more important so that similar systems and data can be implemented on different platforms that may exist across the organization. A standard means of identifying, classifying, and specifying system components is also required. Interface standards embodied in frameworks such as those outlined in the CIM Technical Reference Model can form the basis for interface and component specifications.

We will identify opportunities for cross-functional systems and implement systems in such a way that we can take advantage of standard components throughout the organization.

Agree ← **Implications** → *Disagree*

- Need to identify generic components and how they are implemented
- Standard application and data definitions are critical
- Planning, architecture, and development process needs to incorporate cross-functional review
- Significant reuse of design, code possible, but significant change in IT process, incentives and culture required
- Role for a "repository" of standard elements and their definition

- Organization likely has a strong line-of-business orientation with significant business unit autonomy
- Potential problems with consolidation of information across organization
- View that similar functions contain enough differences that reuse of standard components would not be beneficial (too much modification)

Industry standards

An organization's position with regard to the source and use of standards is a critical factor in its position with regard to open and proprietary systems.

Organizations that have completed a principles definition process typically become favorably disposed toward using industry standards, especially if they have an orientation toward reuse of system components and cross-functional systems. The perceived risk of continuing to be vendor dependent is too high making the shift toward more open, industry standards appear less risky in the long run.

Our standards and technology choices will be based on vendor neutral standards where available and implementable.

Implications

Agree ← → *Disagree*

Agree	*Disagree*
• Role for open systems standards and technologies	• Perception that industry standards are not mature enough for use
• Development of standards and their implementation by vendors needs to be carefully tracked	• Focus more on a vendor's product architecture—vendor-dependent such as AS, NAS, etc.
• Migration strategies need to be developed for utilizing industry standards	• Organization unable to deal with migration issue at present time or does not see benefit of migration
• Focus on standards selection, then technology selection to support standards	• Limited set of vendors
• Need a mechanism for evaluating products in terms of compliance to standards and how to select products where standards have not been defined	
• Evaluate organization's industry standards as well as IT industry standards needed	

Measurement

While at first the issue of measurement would appear to be self-evident, the organization's attitude and investment in measurement and metrics vary dramatically. Some organizations view IT as delivering substantial business value—the actual IT measurements may not be critical. Other organizations, especially ones focused on IT efficiency, may want to have explicit metrics of many facets of the IT environment and its impact on the organization.

There are many measurement areas revolving around IT productivity, efficiency, and quality. Some statement of the organization's belief about measurement needs to be stated, as it will affect management processes around justification and direct investment in measurement programs.

Applications and technology components (processors, network, etc.) need to be implemented in such a way that measurement data can be captured for analysis and management of the IT environment.

Implications

Agree ← → *Disagree*

• Appropriate metrics and explicit indicators need to be established as well as a management process for collecting and managing the measurement information	• View that IT provides intrinsic business benefits and the added cost for implementing measurement capabilities are not justified
• Technology components need to provide data about their operation and performance	

Efficiency vs. effectiveness

Whether to focus on IT effectiveness or IT efficiency is closely tied with the organization's view of measurement. IT effectiveness tends to focus on external measures, ones that are often hard to quantify; these include evaluating the business value of implemented systems, the impact IT has on the organization's market share, etc. Efficiency, on the other hand, focuses more on internal measures such as productivity, cost control, processing efficiency, and transaction costs. The organization's belief on this issue can indicate their view of IT. Is IT a needed but unwanted expense, or is IT critical to the organization's success in its mission?

Information technology has a critical impact on our organization's business success. We must focus on improving IT's impact on operations.

Implications

Agree ← → *Disagree*

• IT viewed as a strategic asset	• IT is viewed as a support organization and not necessarily critical to the organization's mission
• External effectiveness measures critical	• Internal efficiency and cost control measures key
• Value focus—emphasis on increasing business benefits of IT	• Cost focus—increased cost and downsizing pressure on IT
• Probably more willing to take risks on IT investments	• Risk adverse

Security

A statement on security and contingency planning is needed if roles and responsibilities are not defined or the policy is unclear. While the need for secure systems can be considered "motherhood," the organization's view of where and how security is implemented is important to state.

The following principle is one example of a security principle:

Implementation of security measures and contingency plans are the responsibility of the business unit manager where the system is implemented and must be "orange book" compliant.

Implications

Agree ←	→ *Disagree*
• Decentralized approach to security	• More centralization
• User managed security—IT plays an advisory role	• IT is responsible for security
• Process needed to ensure adequate security and contingency plans are implemented by the business units	• Security can only be achieved through central control

IT organization

The IT organization principles deal with the organization's view of how IT is organized and how it interacts with the business. These principles will have an impact on the degree of dispersion of IT and the role of a centralized (if any) IT organization.

Dispersion

Dispersion deals with the degree of control and autonomy that business units have over IT decisions. In a highly dispersed organization, business units make essentially all the IT decisions and may implement systems. In a non-dispersed organization, most IT-related decisions are made within the IT function.

Dispersion is different from centralization/decentralization. It is possible to have a decentralized IT function that is not dispersed. In this case, individual units may have their own IT functions. On the other hand, in a centralized IT function with dispersed control, the IT function may provide resources or advice to the business units.

Our IT organization will become increasingly dispersed into the operational units but policy will be made centrally.

Implications

Agree ←	→ *Disagree*
• Increasing control will migrate to the business units	• Centralized management of IT
• Diminishing role and control for central IT	• Linkages with business units have to be defined and managed
• Standards become critical to ensure that data can be shared and to limit potential duplication of effort across the organization	• Standards can be managed centrally
• Focus on "strategic" systems that directly support the business unit	
• Technology decentralization needs to be addressed—likely increased pressure for distributed computing	

System ownership

Management and ownership of implemented systems and technologies must be addressed to clarify roles and responsibilities. This principle has a direct impact on the rights and obligations of the business unit and IT managers.

Successful implementation and operation of information systems and technology is the responsibility of the business and operational unit(s) that the system supports.

Implications

Agree ←	→ *Disagree*
• Responsibility for systems implementation is the business unit manager's	• IT likely responsible for successful implementation and realization of benefits
• Business unit managers must understand how to successfully utilize IT and manage implementation projects	

Role of centralized organization

The role of the centralized organization is closely related to the organization's view of IT dispersion. In a highly dispersed environment, centralized IT may be responsible only for corporate financial and reporting systems and some standards setting (as in the following example). In other cases, the centralized IT organization may be a source of resources for business unit projects or, in the centralized case, be responsible for all IT functions.

The centralized IT organization will ensure that systems comply with our organization's standards and will assist organizations in IT.

Implications

Agree ←	→ *Disagree*
• Standards setting, advisory, and compliance verification role	• Implies either a limited role for centralized IT or a highly centralized IT function
• Management processes needed to develop and promulgate standards	• Standards and standards compliance performed by centralized IT organization

Life-cycle management

Development, change management, and retirement of systems and technology infrastructure need to be managed on an ongoing basis. Architecture, and the systems developed from the architecture, must take into account constantly changing business needs and evolve with the business.

Our systems should be developed in such a way that they recognize the need for future changes to functional and technology requirements even if the development cost is increased.

Implications

Agree ←	→ *Disagree*
• Willing to accept additional cost to build and achieve easier maintenance and change management	• Limited change to technology
• Acceptance of changing technology may imply need for portability and portable environments	• Perceived to have stable technology base
• Increasingly shared and integrated systems will require adequate change-management facilities	• Standards less important
• Desire reduced maintenance cost and time	
• Increasingly modular design to facilitate change	

Application management

Application management principles deal with the organization's stated directions for managing applications and application components.

Depending upon the context for the architecture, these principles can focus on structural system issues (portability, modularity, etc.), management issues (methods, techniques, distribution), or some combination of the two.

Taken as a whole, the application management principles have to state the organization's beliefs on "How will we distribute and manage applications to get the maximum benefit for the organization?"

Development process and methods

The role of a development methodology and associated techniques across an organization should be addressed.

Systems should be implemented using a consistent methodology across the organization.

Implications

Agree ←	→ *Disagree*
• Common development methodology • Promotes sharing of techniques and language	• Methodology managed by individual business units or departments

Reusability

The issue of reusability of applications and application components is analogous to many of the data standardization efforts under way. Standard definitions of business functions and application components are addressed in the following principle. This can be used to expand the focus of reusability beyond sharing code to sharing business designs, documentation, etc. Potentially, investment could focus more on an expanded system repository or I-CASE tools.

Applications should be developed using standard system components that are shared across the organization

Implications

Agree ←	→ *Disagree*
• Standard definitions	• High degree of customization required (possibly by business)
• Library of shared components needed	• Limited cross-functional systems
• Components may be source code, application designs, documentation, etc.	
• Migration to this environment needs to be planned—promote activities that provide shareability	

Build or purchase

The make-versus-buy issue needs to be resolved. Organizations can swing either way on this principle, depending on their view of the uniqueness of their business or applications.

Where possible we will purchase systems and components of systems by using commercial off-the-shelf (COTS) products

Implications

Agree ←	→ *Disagree*
• Buy orientation	• Tendency to build applications
• Procurement standards and process need to be defined	• Development resources and standards critical
• Cost of modification needs to be incorporated into package cost	

Cross-functional opportunities

Many organizations today have missed the opportunity to reuse portions of applications (see standard components principle above) by not identifying and architecting systems that support similar functions in multiple areas.

This represents a significant opportunity to improve productivity and obtain some economies of scale through functional or technical integration.

Applications should be developed so that they can be reused in similar business functions across the organization

Implications

Agree	Disagree
• Process needed to determine what portions of an application are generic versus specific to the particular business function	• Traditional design methods adequate
• Standard interfaces and standard components needed	

Distribution of application functions

Distributing application functions away from centralized data centers will have a significant impact on the resulting architectures and management processes required to manage a distributed applications environment.

Application systems should be distributed and executed as closely as possible to the users of the application.

Implications

Agree	Disagree
• Decentralized, workstation focus	• Centralized, host-processor focused
• Architecture levels (personal, work group, enterprise) need to be defined consistent with the organizational levels	• Traditional application design and development environment
• Potential cooperative processing implementation	• Economy of scale
• Potential multivendor implementations of similar software	• Likely single or limited set of vendor platforms
• Critical need for application environment standards	

Common application environments

A principle such as the one stated below supports a vendor-independent, portable environment. The result is a strong focus toward open systems.

Applications should be developed in a common environment that is independent of the underlying technology.

Implications

Agree ←	→ *Disagree*
• Open systems needs to play a key role	• Common environment may be single vendor
• Multivendor, standards based—standards are key	• Open standards less important than vendor products and standards
• Network computing standards required	
• Architecture models must deal with creating an "opaque" layer between application and technology	

Common user interface

The need for a common user interface (one with the same behavior) has emerged as an important requirement in many organizations. Common user interfaces (CUIs) can potentially provide significant improvements in productivity and training for users.

In the context of this principle, a common user interface does not necessarily imply a *graphic* user interface though the two seem to becoming synonymous with each other. The ability to customize the CUI for a particular need is often important as a generic CUI may not provide the best solution in all cases. The issue of migration from existing character terminals will need to be addressed, especially if graphic user interfaces are the chosen direction.

Applications should present a common user interface that is adaptable and extendible to particular user requirements.

Implications

Agree ←	→ *Disagree*
• Selection of an extendible common user interface toolkit critical	• Application-specific interfaces
• Migration from existing workstation and terminal technologies need to be addressed	• Easier migration—can utilize existing installed base
• Application design standards and procurement standards need to incorporate CUI standard	• Purchasing software and hardware not restricted by CUI
• Multivendor workstations and PCs may need to be accommodated	• Able to customize user interface to specific applications
• Can integrate applications from various sources more easily (from user perspective)	

Information management

The organization's approach to managing information is addressed in the information management principles.

Multiform vs. single form

The scope of the information managed and the degree of integration of different forms of information are addressed in the following sample principle.

Our architecture and implemented systems must address the management of all forms of information (data, text, voice, image) in an integrated manner.

Implications

Agree ←	→ *Disagree*
• Compound document standards required	• Easier to implement today
• Information architecture needs to address all forms	• Oriented toward data and possibly text management
• Tools and standards needed	• Standards available

Data standardization

The organization's view on data standardization needs to be articulated. Is there a need for standard definitions? Is the expense and effort justified? Is the organization so decentralized that standardization efforts are not really worthwhile?

Standardization of data definitions and their implementation, access, interoperability, and communication is needed across the organization to provide improved quality and consistency of data and improve overall effectiveness of implemented systems.

Implications

Agree ←	→ *Disagree*
• Standard definitions required	• Limited communication across organization
• Significant effort and cost associated with standardization effort	• Limited need or ability to consolidate data
• Improvement in sharing and consolidation of data	

Ownership and stewardship

Ownership and stewardship of data needs to be addressed and agreed upon. This principle has a number of implications on how roles and responsibilities get defined.

Data is a corporate asset and does not belong to a particular business unit or individual.

Implications

Agree ←	→ *Disagree*
• "Corporate" ownership and management of data	• Business unit management and control of data
• Management of data may be delegated—process is required to ensure it is managed correctly	• Fragmentation of data a possibility—standards for consolidation and interchange required

Data access and distribution

Like distribution of application function, distribution of data should be resolved. The organization's view on application and data distribution helps determine its "style" of computing—centralized, decentralized, or some combination of the two.

Data will be managed and stored as closely as possible to the person/organizational unit who uses it.

Implications

Agree ←	→ Disagree
• Distributed data management required	• Data can be managed using existing technologies
• Degree of sharing needs to be established for data	• Accessibility of data by distributed users needs to be addressed

Technology management

A wide variety of technology management principles can be stated by an organization. Organizations often break such principles down into key topic areas that deal with the major components of the technology architecture (hardware, system software, communications, etc.).

It is important to articulate the role of each technology component and the organization's attitude toward managing vendors and technology. With the input from the application and information management principles, the technology management principles are where the organization's position toward open systems and standards often gets stated in black and white.

Interchangeable components

The first principle addresses interchangeability of vendor products and services and, by implication, the organization's view toward open systems and standards.

Some organizations view this principle (as stated) as unachievable in a reasonable time frame and choose a more conservative approach.

We will implement technology components so that we have the option of exchanging vendor products with minimal disruption to the environment.

Implications

Agree ←	→ *Disagree*
• Standards-based, open-systems approach	• Lock-in to a vendor
• Interfaces and environments need to be standardized	• Can stay with existing technology base or do a selective migration.

Vendor management

An alternative statement of the following principle could be:

"We will limit the number of alternative vendors to a limited, manageable set" or *"We are committed to a single-vendor environment."*

We will utilize any vendor who provides us with the best technology for a business need.

Implications

Agree ←	→ *Disagree*
• Need for standard environments to support multivendor	• Allowable vendor set needs to be established that can meet most needs
• Standard connectivity approaches needed	• Limited set of vendors can be managed—build stronger relationships
• Portability of applications and data must be addressed	

Distribution of processing capability

We will decentralize our processing environment so that individual business units control their own computing resources.

Agree ← **Implications** → *Disagree*

• Promotes highly distributed environment	• Promotes more centralized environment
• Remote management and standardization critical	

Role of intelligent workstations

The following principle brings intelligent workstations (PCs, workstations) to the forefront as a platform for delivering applications in the architecture.

The result is a highly distributed, processing environment. Applications and access to data would be provided through the workstation, supplemented by servers and hosts (minicomputers and/or mainframes) providing processing and data services to the workstations.

Intelligent workstations will be the primary access and delivery vehicles for applications and data.

Agree ← **Implications** → *Disagree*

• Workstation, LAN orientation	• Delivery through minis, mainframes
• Workstation standards and connectivity critical—network computing needed to integrate with other components	• Standards critical, but applications and data less distributed
• Strong network computing role	

Network connectivity

The following three principles address networking issues. The first establishes the role of the common network utility.

We will use a common network environment using industry standards to interconnect all workstations, computers, and communicating devices.

Implications

Agree ←	→ Disagree
• Connectivity standards need to be defined	• Autonomous computing/network environments
• Full set of communications and transport facilities will be required	• Tend to stay with host-based communications—terminal to host
• Vendors need to support interconnectivity	• Can implement different networking approaches in different areas of organization
• Integrated LAN/WAN/external network design required	
• Connectivity with customers/suppliers needs to be addressed	
• No system is an "island"	

Network interfaces

The following principle supports the premise that the "network is the computer" by placing the common network environment as the core through which all devices communicate. Standard protocols and interfaces begin to establish the need for a common interface standard.

The OSI model is often used to describe the various interface layers and as a framework for identifying standards.

All communicating devices must interface to the network through a standard set of protocols and interfaces.

Implications

Agree ←	→ Disagree
• Limit direct connection of devices to computers—connect through common network	• Point-to-point links (e.g., computer to workstation) allowed
• Migration from existing installed base needs to be addressed	

Network services

The network's role as a value-added service provider is established in the following principle. This represents a belief that value-added services can be delivered by the network separately from the processors attached to the network.

Common services such as file transfer, electronic mail, directory management, and network management should be provided through a common networking environment.

Implications

Agree ←	→ *Disagree*
• Value-added applications need to be provided	• Common services provided by host processor
• Security of directories and services need to be addressed	• May need to integrate different services at the workstation instead of through the network
• Additional system management services should be examined	
• Network-based processors for network services need to be defined	

Conclusion

The above-listed principles are but a few of the many that an organization may seek to develop. We recommend that all the existent CIM principles be incorporated into each architecture effort.

Appendix B: How To Do A Baseline Characterization

General approach

The baseline data collection effort is the first step in developing a useful baseline characterization of the current architecture. Standard templates were developed over the course of the first SBA projects that were completed (or are under way) at the time this update to the SBA Guide was produced.

The first section of this appendix presents these templates, along with the instructions that accompany them, and includes a sample of a completed template from the USMC project.

The second major section of this appendix provides guidance for the analysis of the information generated from the completed templates. General questions of interest and "rules of thumb" for analysts are provided.

This page intentionally left blank.

Baseline Data Collection

Work Organization Templates

This page intentionally left blank.

Business and Work Models Template

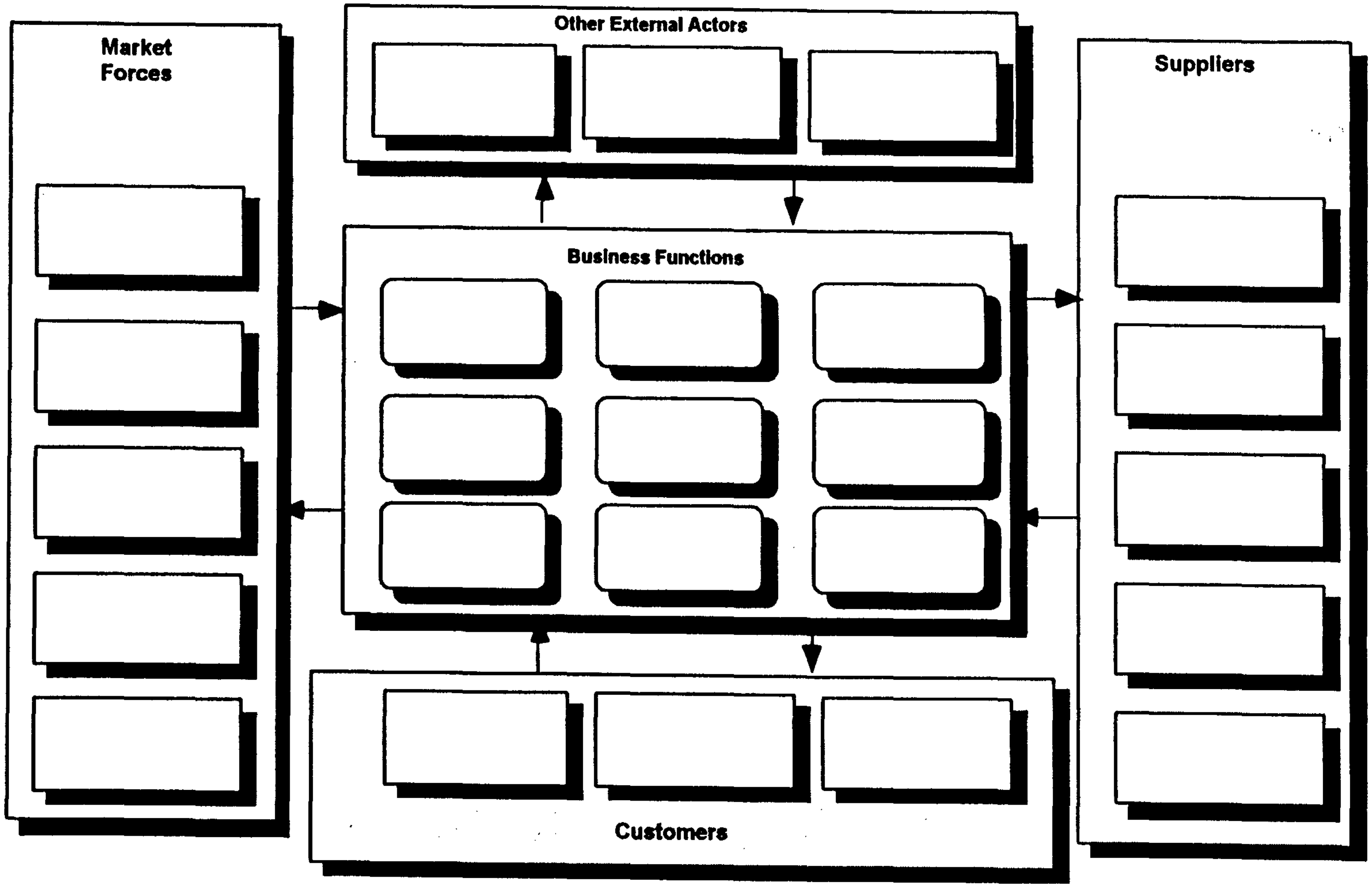
- ***Fill out one of these templates for each major business function in the enterprise***
- ***See the Baseline Assessment Glossary of Terms for definitions***

Mission	• *This is the organization's mission:*
Function	• *A major grouping of work for the enterprise:*
Processes	• *Activities or job steps leading to a desired result within the function:*
Location	• *Physical location(s) where work is performed:*
Headcount	*This should also be entered on Baseline Template for Function Costs.*
Budget	*This should also be entered on Baseline Template for Function Costs.*

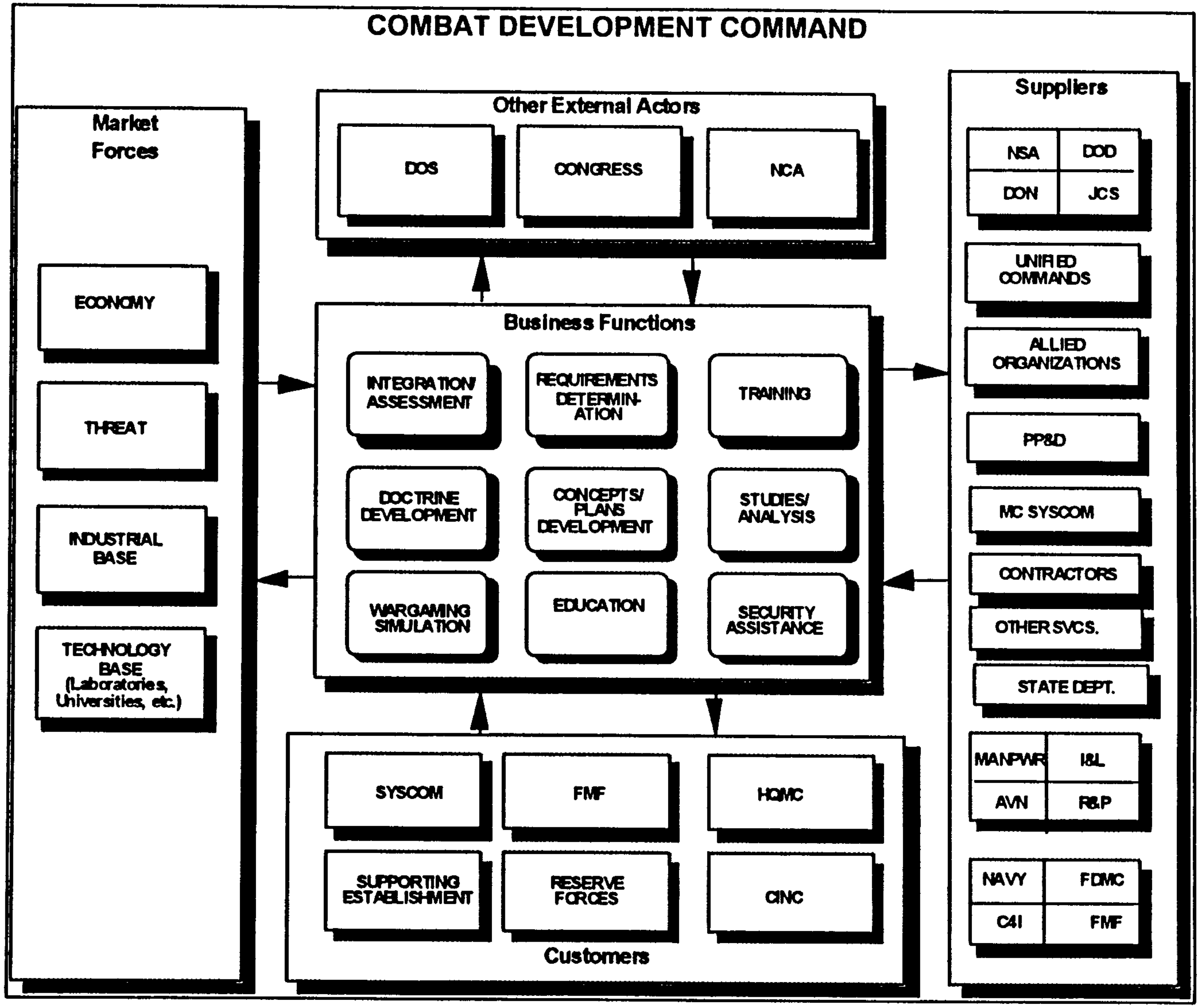
Business Context Template

Competed by: ____________________

- ***Fill in name of organization or business Unit:*** ____________________
- ***Then fill in the boxes with appropriate information about your organization or business unit***



Business Context - Sample



This page intentionally left blank.

Information Templates

This page intentionally left blank.

Baseline Template – Application to Information

Existing Application(s) Linked to Data Groups:

- *List all known existing applications which support business functions within your organization in the left-most column of this matrix. (give common application abrreviation and full name if known)*
- *List all known data groups across the top of the matrix, one per column. Attempt to list closely related data groups in adjoining columns under a larger heading called "Subject", i.e. the Subject will span one or more data group columns.*
- *If more columns are needed repeat the applications on a second sheet and continue listing the domains and information subjects.*
- *Place a C, R, U, and/or D in the intersection of the application and information subject to signify whether the application Creates, Reads, Updates, and/or Deletes the information subject as part of its functionality. Combinations are Possible.*

This template should contain all data groupings and applications of the entire enterprise.

Subject													
Data Group Application													

Baseline Template – Application to Information – Sample

Subject / Data Grouping / Application	Customers			Finished Products & Services			Supplier			Transport		Plant Equipment & Facilities	
	Customer Information	Business Agreement	Orders	Products	Inventory	Services	Supplier Information	Business Agreement	Purchase Order	Transport Acquisition	Transport Utilization	Equipment	Facilities
Planning Applications													
New Product Concept and Planning System	R			CRUD		R						CRUD	R
Executive Information System	R	R	R										
Profit Planning System	R	R	R	R	R	R	R	R	R	R	R	R	R
Stategic Planning System	R	R	R	R	R	R	R	R	R	R	R	R	R
Profitability Analysis and Reporting System	R	R	R	R	R	R	R	R	R	R	R	R	R
Market Analysis and Trending System	R	R		R		R							
Capital Planning and Tracking System	R	R	R	R	R	R	R	R	R	R	R	R	R
Patent & License System				CRUD		CRUD	R	R				CRUD	CRUD
Selling Applications													
Customer Business Agreement System	CRUD	CRUD	R	R		R				R		R	R
Sales Demand Forecasting System	CRUD	R	R	R		R							
Call Reporting System	CRUD	CRUD	R	UD		UD							
Contract Versus Actual Reporting System	CRUD	CRUD	R	R		R							
Product Applications Technology System	CRUD	R		R		CRUD						R	R
Consulting Service Tracking and Problem Resolution System	CRUD	R	R	R		CRUD						R	R
Advertising and Promotion Scheduling and Information System	R			R		R							
Credit Management System	CRUD	R	R				R	R	CRUD				
Product and Services Delivery Applications													
Order Management System	CRUD	R	CRUD	R	R						R		
Complaint Tracking and Resolution System	CRUD	R	CRUD	R	R						R		
Customer Information System	CRUD	R	R	R		R				R		R	R
Finished Product Inventory System			R	R	CRUD							R	CRUD
Production Scheduling System			R	R	R							R	CRUD
Automated Load Out System	R		CRUD	R	R						CRUD		
Transport Scheduling and Optimization System	R		R	R							CRUD		
Fleet Maintenance and Inventory System	R	R		R			R	R	R	CRUD	CRUD	CRUD	
Fleet Acquisition System	R	R		R			R	R	R	CRUD	R	CRUD	

Database Inventory – Template

Instructions:

Fill in the requested information for databases and/or electronic files which support your area of operation. The following definitions of the columns apply:

Database Name The commonly used name or acronym for the database or file (i.e. SASSY, MIMMS, TMS, etc.)

Description A brief description of the contents of the database or file (i.e. personnel records, patient records, requisitions, etc.)

Geographic Location The location where the database or file physically resides (i.e. the commonly accepted designation, such as city, base name, command name, etc.)

Type The type of database or file system used (i.e. DOS file, dBase, Adabas, Oracle, Paradox, DB2, etc.)

Platform The type of computer platform (i.e. PC, Workstation, Midrange, or Mainframe;
also state specific kind and model number if known, such as IBM PS/2, Sun Sparc Station, AS/400, DEC/VAX, IBM 3090, etc.)

Additional Notes Any other clarifying notes which your feel will be helpful in characterizing the Database or File

Database Name	**Description**	**Geographic Location(s)**	**Type**	**Platform**	**Additional Notes**

Database Inventory – Sample

Database Name	Description	Geographic Location(s)	Type	Platform	Additional Notes
3M	Maintenance, Scheduling, and Parts	Philadelphia ASO		Mainframe NAVY	
BNA	*Training/Assignment*	Quantico CDPA	Adabas	Mainframe	
CAEMS	Embarkation Data (size, weight, bal, etc.)	Numerous Commands	Paradox	PC	
CAIMS	Ammunition (AVN)	Philadelphia ASO		Secure Dial-in to Mainframe	
Central DB System	MTF Formats, Data Codes, Syntax Checks (also TADIL formats)	Reston JIEO		Sun Sparcstation and 7	
Control Master File	Personnel and Pay	Kansas City CDPA	Adabas	Mainframe	
Defense Intelligence Data Sys	Counter Terrorist/Counter Intelligence	Bowling AFB DIA	RDB	DEC VAX	
Emerald	Counter Narcotics	Bowling AFB DIA	Sybase	Sun (client-server)	
HAS	Accounting	Quantico	Adabas	Mainframe	
IAS Database	Portion of MIIDS/IDB plus Tactical Update	MEF, DIV, REGT(?)	Sybase	Sun Sparcstation 2 or 1.0	BN version working at Army LCU Laptop
LFADS	Supply and Equipment Data	Numerous Commands	Clipper/ADA	PC	
MAGTF Data Library	Reformatted Data of all types	Major Commands	Clipper	PC	
MAGTF II	Org and Transport Data	Numerous Commands	Clipper	PC	
MCAIMS	Student Information/Course Data (Central File at Quantico planned)	Each School	Adabas	PC	
MCCRES	Unit Scores, Mission and Performance Standards	Quantico	Adasage	PC	
MCLLS	Lessons Learned Notes (distributed by CD to Major Commands)	Quantico	dBase/Clipper	PC	
Met Table	Meteorological Data	Each BCS		PC	
MIIDS/IDB	All Non-US Military Info, Electronic Order of Battle, Airfields/Facilities, General Military Intelligence	Bowling AFB DIA	Model 204	Mainframe	
MIMMS	Maintenance Data	Numerous Commands		Mainframe	
NALCOMIS	Maintenance/Supply Data	Numerous Commands	Cobol	Mainframe and PC	
NALISS	Supply Parts	Philadelphia ASO		Mainframe	
NAVFLIRS	Extract of Flight Info/Pilot Info	Norfolk NAVNASSO		PC to Mainframe Tape	
Ord Table	Ordnance Characteristics	Each BCS		PC	
Org, Equip, and Supply Data	Org, Equipment and Supply Data	Numerous Commands	Paradox	PC	
POM	Fiscal Data				
Resource Allocation Display	Navy/Marine Corps Shared Financial Data	Washington Navy HQ Op-80		Mainframe	
SABRS	Accounting and Budget	Quantico CDPD	Adabas	Mainframe	
SASSY	Parts Data	Albany/Barstow		Mainframe	
SUADPS	Financial and Inventory Data	Numerous Commands	Cobol	Mainframe	
TCAMS	Travel Data	Numerous Commands	Clipper	PC	
TDMS	Parts Technical Data		Adabas	Mainframe	
TERPES	MIIS/IDB EDB Plus MISSLE OB, GOLDDB and Tactical copy of updates (send tactical version back for updates to MIIDS/IDB)	VMAQ2 (four locations)	Sybase	Sun 620 Fileserver	
TMR Data Base	Organization Structure and History	Quantico CDPA	Adabas (w/CICS)	Mainframe	

Application Templates

This page intentionally left blank.

Application Inventory – Template

Instructions:

Fill in the requested information for all applications which support your area of operation. The acronyms used on this template should also be used on any other templates which refer to application systems. The following defintion of columns apply:

Application Acronym The commonly used abbreviation for the application

Application Name The full English name of the application

Type Denote whether On-line, Batch, or Both On-line and Batch ("O", "B", or "OB")

Number Users The number of users

Language The language(s) the application is written in

Operating System The Operating System and On-line Transaction Processing Monitor (if applicable) under which the application runs

Where Run: Specific Technology Platform The commonly used designation for the technology platform upon which this application runs (i.e. the unique name for a particular processor and the basic vendor model information, such as "DAISY2, IBM3090 Model 600E")

Where Run: Location of Platform The actual locations where the application runs on the platform identified in the prior column

Age The age of the application in years

Number Programs Number of executable programs in the application

Changes Requested Number of change requests in past year; over entire history

Changes Implemented Number of change requests implemented in past year; over entire history

Failures Number of application failures in past year; over entire history

Developer What organization developed the application

Support What organization supports the application

Application Acronym	*Application Name*	*Type*	*Number Users*	*Language*	*Operating System*	*Where Run*		*Age*	*Number Programs*	*Changes Requested*	*Changes Implemented*	*Failures*	*Developer*	*Support*
						Specific Technology Platform	*Location of Platform*							

Application Inventory – Sample

Application Acronym	Application Name	Type	Number Users	Language	Operating System	Where Run		Age	Number Programs	Changes Requested	Changes Implemented	Failures	Developer	Support
						Specific Technology Platform	Location of Platform							
AB	Automated Budget													
ABA	Bonds & Allotments		All MC	COBOL II	MVS-XA	IBM, AMDAHL, BURROUGHS, UNISYS, TANDEM	LEJEUNE, PENDLETON, OKINAWA, EL TORO, CHERRY PT , KANSAS CITY	22	159	27	27		KANSAS	KANSAS
ABA	Bonds & Allotments		All MC	COBOL II	MVS-XA	IBM, AMDAHL, BURROUGHS, UNISYS, TANDEM	LEJEUNE, PENDLETON, OKINAWA, EL TORO, CHERRY PT , KANSAS CITY	22	159	27	27		KANSAS	KANSAS
ABE	Application B Enhanced			TACL	GUARDIAN	TANDEM	CHERRY PT	5	68				FMSO	FMSO
ACIS	Automated Claims Information System		MH Claims	DBASE III	MS-DOS	PC	HQMC	5	21					CONTRACT
AFRS	Automated Fitness Report System		MMPF	NATURAL	MVS-XA	IBM	HQMC	20	216	270	270	0	QUANTICO	QUANTICO
AIMS	Awards Information Management System		MH	FORTRAN	MS-DOS	PC	HQMC	4	200					CONTRACT
ALPS	Automated Leave and Pay System		CIV PAY OFFICE, NAVCOM P, IRS	COBOL	MVS-XA	IBM, AMDAHL, BURROUGHS, UNISYS, TANDEM	QUANTICO, ALBANY, KANSAS, LEJEUNE, PENDLETON, OKINAWA, EL TORO, CHERRY PT	23	85	92	89	1	QUANTICO	QUANTICO
AMHS	Automated Message Handling System			C/TOPIC	UNIX	DPC	MEF, Garrison, CPs	NEW					DIA	
AMMOLOGS	Ammunitions Logistics System			COBOL	MVS-XA	IBM, AMDAHL, BURROUGHS, UNISYS, TANDEM	LEJEUNE, PENDLETON, OKINAWA, EL TORO, CHERRY PT , KINSER PARIS ISLAND	4	12	1	1	6	ALBANY	ALBANY
AN/TPS-59	Radar Set 3D	OL	13 Units	CMS-2, ULTRA-32		AN/UYK-7	AIRWING	7						
AOWP	Automated Orders Writing Process		MMEA, MMOA, MMSR	COBOL II	MVS-XA	IBM	QUANTICO, KANSASNCITY	13	55	33	31	3	KANSAS	KANSAS
APADE	Auto procurement			TACL	GUARDIAN	TANDEM	CHERRY PT.	3	1458				FMSO	FMSO
APCS	Automated Production Control System							NEW						
ARMS	Automated Recruit Management System		Recruiting	COBOL II	MVS-XA	AMDAHL	KANSAS CITY	13	675	11	11	0	KANSAS	KANSAS
ASMS	Advanced Shipping and Monitoring System			TACL	GUARDIAN	TANDEM	CHERRY PT.	3	10				FMSO	FMSO
ATAC+							(Maintained by the Navy)							
ATRIMS	Aviation Training & Readiness Information Management System		ALL FMF	ADA	MVS-XA	IBM	MC WIDE	8	1				QUANTICO	QUANTICO
AV-3M/NAVFLIRS	Aviation Maintenance Material Management System Naval Flight Record		Aviation Units	COBOL II, ADA	MVS-XA, MS-DOS	IBM, AMDAHL, BURROUGHS, UNISYS, TANDEM, PC	LEJEUNE, PENDLETON, OKINAWA, EL TORO, CHERRY PT.	6	72	41	40	2	QUANTICO	QUANTICO

Appendix H: How To Do SBA Administration

SBA administration

Most organizations recognize the "need for SBA governance" on or about the time that the initial SBA planning project comes to a close. It is strongly recommended that the DoD adopt a mechanism for keeping the SBA up to date.

It is not uncommon for an organization to establish an SBA administration function that coordinates the review of SBA-related projects and resets project priorities based on architecture evolution.

Typically, this coordination is managed through semi-annual SBA review meetings held with SBA representatives from each of the major functional areas participating in the SBA effort (representatives are selected by the ASC). SBA representative are responsible for keeping the SBA administration function abreast of changes in project status and direction. In turn, the SBA administration uses the representatives to execute changes in the general SBA strategy (consult the *Implementation Plan Document* for more details). The final pages of this SBA Guide describe a recommended process that can be used to support the goals of the SBA.

Process overview

An SBA Management Team (SBAMT) will be established to maintain the SBA. This team will work directly with project managers responsible for developing SBA projects as well as with the functional managers and their staff responsible for overseeing project implementation.

It is paramount that the SBAMT build into the overall administration process a review system to ensure compliance with the objectives set forth in the *Architecture Framework Document, Target Architecture Document, Opportunity Identification Document, Migration Options Document*, and *Implementation Plan Document.*

Monthly project coordination meetings will be held between the SBAMT and all project managers developing SBA-related efforts. The purpose of these reviews will be two-fold:

- Provide an opportunity for project managers to report any issues that will impact the delivery of their projects to the SBAMT, who will approve changes to project plans
- Create an environment whereby SBA project managers can meet to discuss cross-project issues and actively identify opportunities to reuse code and build integrated systems.

On a quarterly basis, the SBAMT will sponsor a status review with the executive sponsor. This quarterly review will provide top decision makers within the organization an opportunity to review the progress of key IT initiatives while lending guidance to the SBAMT.

When the SBAMT is not meeting with project managers or the executive sponsor, they are updating the SBA project plans and communicating all changes to these plans through a myriad of communication vehicles intended to provide needed information to all members of the organization's stakeholder community. (See the "communication vehicles" part of this appendix for more details.)

Key elements of the SBA management process

Following are several important elements in the SBAMT process:

- Establishment of the SBAMT
- Addition of SBA to the duties of the executive sponsor
- Implementation of the project coordination meetings
- Institutionalization of the quarterly SBA reviews.

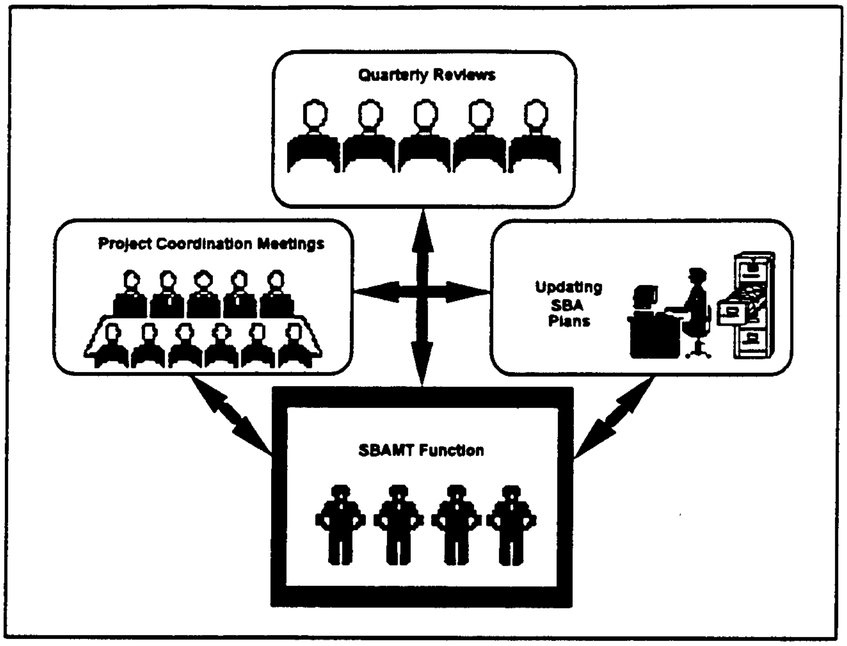


Figure H-1. The SBA Management Team (SBAMT)

The SBAMT

The first step in the SBA administration process is to establish an SBAMT. The SBAMT is charged with keeping the SBA up to date. This is done by managing the coordination of the projects defined in the SBA *Implementation Plan Document*. The people assigned to this function will employ such devices as monthly meetings with SBA project managers as well as quarterly reviews with the executive sponsor in order to ensure that the SBA projects are evolving as planned.

The team should be staffed with experienced planners and technologists who have a deep-rooted understanding of IT implementation projects (i.e., data processing, communications, and systems analysis). Typically, the team is situated in the IT systems development area enabling it to oversee the development activities. If not, standards and policies defined by the SBAMT could be ridiculed because the process "was not invented here." If reorganization occurs, it is important that the SBAMT be placed with the highest ranking IT officer to ensure continued execution of the SBA plans.

Many organizations are beginning to place SBA administration functions under the command of the senior-most executive (i.e., the CEO) in order to ensure that the most crucial IT applications are being developed in unison with the organization's strategic plan. This is highly recommended and represents the best case scenario.

Once established, the team must conduct a general assessment of the SBA projects to see if, in fact, the projects are being implemented in compliance with the overall architecture. This is done by mapping project progress against the implementation plans as well by as the team asking itself (and the responsible project managers) some hard questions like:

- Is the architecture framework still valid? Should any of the architecture principles be modified? Which ones and why? What has changed?
- What are the benefits to be had from changes to the implementation plans? Are there any cost savings, value-added benefits, or softer, long-term intangible benefits?
- Have IT standards been materially implemented in the organization? How far along the standards road have we traveled thus far? How far, given this "process check," do we have yet to go? Have we gleaned 80 percent of the benefit already, or is there still payoff down the road?
- Has the enterprise recognized any benefit from the work achieved?
- Given the current state of implementation, have any other payoffs been obtained that may not have been originally predicted?
- In general, do the plans and their delivery schedules appear to be changing?
- Have any standards, targeted as important, not yet matured as much as originally anticipated?
- What is the status of the technology that was selected for implementation? Has it "shown up on time" in the marketplace? Have we secured its acquisition?

After these questions have been answered, adjustments to the original plans should be made (i.e., if a given project is not maturing as originally scheduled, specific steps must be developed to produce "workarounds").

Primary responsibilities

- Conduct monthly project coordination meetings
- Conduct quarterly executive sponsor meetings
- Update SBA plans
- Communicate SBA changes to the stakeholder community
- Review SBA project status
- Facilitate cross-project sharing of information/code
- Identify opportunities to consolidate systems development efforts
- Assist project managers in adjusting SBA project plans
- Coordinate complimentary voice and data development efforts.

Executive sponsor

In industry, perhaps the largest constraint in SBA implementation work is senior management's unwillingness to participate in the review and nurturing of the IT architecture. To keep SBA in the forefront of activities in the systems development arena, this attitude must change.

An IT steering committee must be formed, charged with overseeing the prioritization of SBA projects, as well as final approval for all changes and adjustments to the SBA project scope and delivery schedules. This duty would be appropriate for the executive sponsor. This team of senior officers should be prepared to commit the necessary resources required to make SBA a success.

Typically, the steering committee (executive sponsor) members participate in quarterly reviews of the SBA project status and actively seek to incorporate input from the quarterly SBA reviews into their budget/planning (i.e., POM) process. These decision makers assist the SBAMT in implementing the necessary changes to the SBA by communicating shifts in priorities to their subordinates.

In this new kind of "top-down," "function-driven" environment, assessment and review become less personally and politically charged. The result is that the SBAMT process becomes easier to conduct successfully. Ultimately, this form of organizational behavior leads to the establishment of a successful and repeatable implementation process.

Primary responsibilities

- Participate in quarterly SBA reviews
- Make decisions regarding SBA project priorities and adjustments
- Oversee SBA project implementation within the functional areas of the enterprise.

Quarterly SBA reviews

Quarterly SBA reviews are a vehicle to help executive sponsor members keep abreast of SBA progress and be aware of all the changes that occur during the SBA project evolution. Information conveyed in these reviews should be incorporated into the budgeting process within the enterprise. In this way, the enterprise will reduce the dollars being squandered on insignificant IT projects.

Also, these reviews are an important means by which the SBAMT can gain an understanding of the desires of senior officers (i.e., balance current priorities with new requirements). This insight will be needed to better manage changes to the SBA project plans and to define new SBA projects.

Primary objectives

- Executive management review of the SBA progress
- Approval and prioritization of new SBA projects
- Approval and prioritization of changes to existing SBA plans
- Providing a means for functional areas to articulate new IT requirements.

Monthly project coordination meetings

Project coordination meetings are held between the SBAMT and all the SBA project managers responsible for building SBA projects. These meetings are a way for the administrators to understand the issues affecting SBA efforts, enabling them to make changes to the SBA.

Furthermore, these meetings are used to encourage project managers to discuss interproject issues, like software reuse and data integration. When this communication vehicle hits its stride, it can be used to deliver information regarding new IT standards and policies to all project managers represented in the coordination meetings.

Primary objectives

- SBAMT review of SBA projects (plans and budgets)
- Announcement of adjustments in SBA plans
- Cross-project discussions on coordination issues (i.e., data sharing, etc.)
- Delivery news on IT related issues (i.e., standards adoption, etc.).

Communication vehicles

As mentioned earlier, it is extremely important to staff the SBAMT with seasoned IT professionals. To do otherwise can be disastrous. Team members must come to the planning table with experience in technology planning and the sensibilities to understand the inherent cultural and political climate.

The next most important factor in conducting successful architecture administration is the establishment of a set of effective communication mechanisms that can help the administration team distribute important information, such as project planning documents, and receive critical feedback without having to become immersed in the typical "red tape" that such work usually entails.

Figure H-2 highlights this issue and suggests several ways the Marine Corps can keep the communication lines open while effectively distributing valuable information about the status of its SBA projects.

Quality review meetings

Sometime during the first year of SBA administration, the SBAMT should develop a quality review process that will be applied to each SBA project as it matures through the phases of the project development life cycle. This "process check" should conform to existing Total Quality Management (TQM) initiatives and, as such, provide a "quality assurance" dimension to the overall architecture administration process.

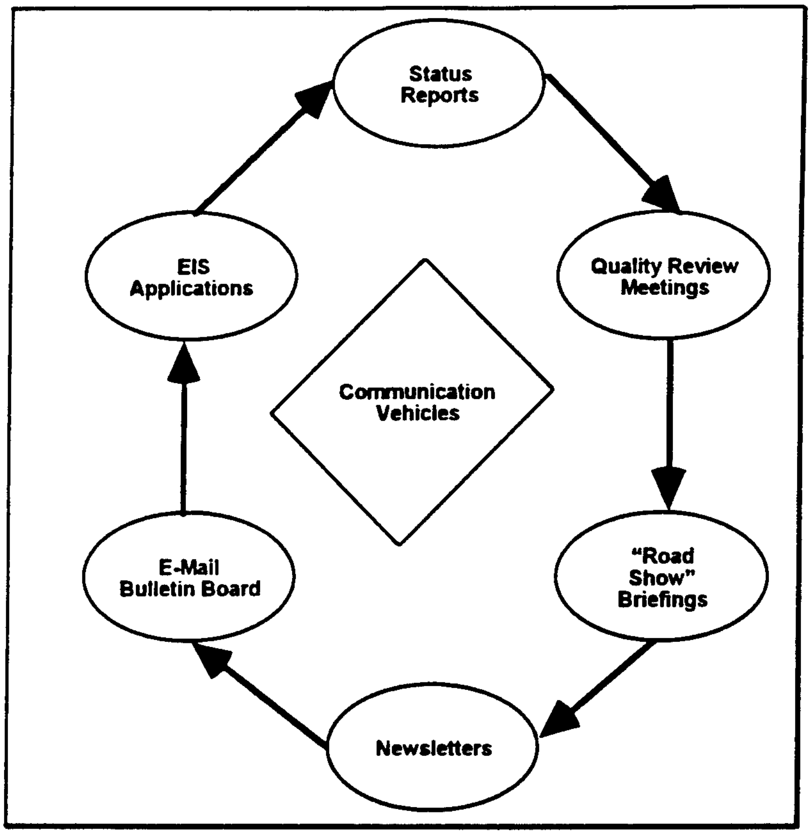


Figure H-2. Some Important Communication Vehicles

A review process based on the Continuous Process Improvement Cycle (see Figure 8-1) is recommended. The notion is that a project is planned, work begins, the result is checked against the plan, and opportunities for improvement are defined and acted upon through modifications to the next plan (or project phase, whatever the case may be). The use of this technique will help the enterprise learn from its SBA experiences.

Each review meeting can be used as a way for the SBAMT to communicate suggested changes in the project development process to SBA project managers (internal as well as external personnel), contributing to the creation of the "learning organization," which is fundamental to TQM objectives.

Status reports

Status reports are another way to improve communication within the SBA development environment. By documenting such things as causes of project delays or scope changes, the SBAMT can begin to define ways to proactively address them. These "lessons learned," together with the modified plans, should be included in a quarterly SBA status report and delivered to all designated personnel.

Often overlooked, documenting the "lessons learned" (see Figure 8-4) becomes very valuable to future project development teams, particularly when defining modifications to SBA project plans helping future project managers to "never make the same mistake twice."

"Road shows"

Another important way to inform enterprise personnel about the significance of SBA is to establish an SBA awareness program (or "road show"). The road show will involve the creation of an SBA briefing that describes the SBA process and explains the impact it has on the enterprise. (See Figure H-3.)

The SBAMT will schedule briefings at all major sites. All personnel would be expected to attend one of these briefings. Once all personnel have been exposed to the SBA project, the next phase of the awareness program would take the form of annual status meetings delivered at the same sites.

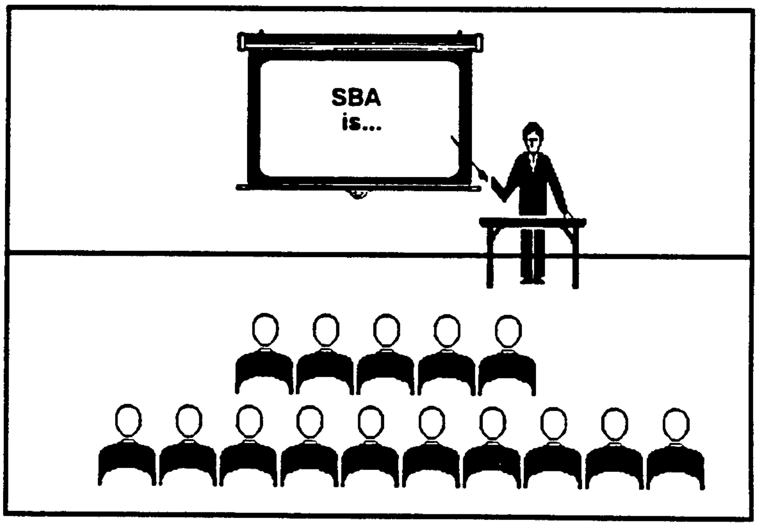


Figure H-3. The SBA "Road Show" Will Take the Message to the Troops

Newsletters

An SBA newsletter could also be created as a means of keeping all personnel informed of the SBA progress. The newsletter could be published quarterly, and its production should coincide with the IT executive sponsor meetings. This way, news concerning executive management decisions about SBA events can be delivered to the entire community.

Electronic bulletin boards

An electronic bulletin board dealing with SBA subjects can be established within the E-mail environment. (See Figure H-4.) It can become a very useful broadcast mechanism, since many personnel use it on a daily basis. In fact, many organizations in the commercial world use such devices as a way to solicit improvement ideas from personnel, transmit newsletters, distribute results from quarterly reviews, and deliver project progress reports to SBAMT-like groups.

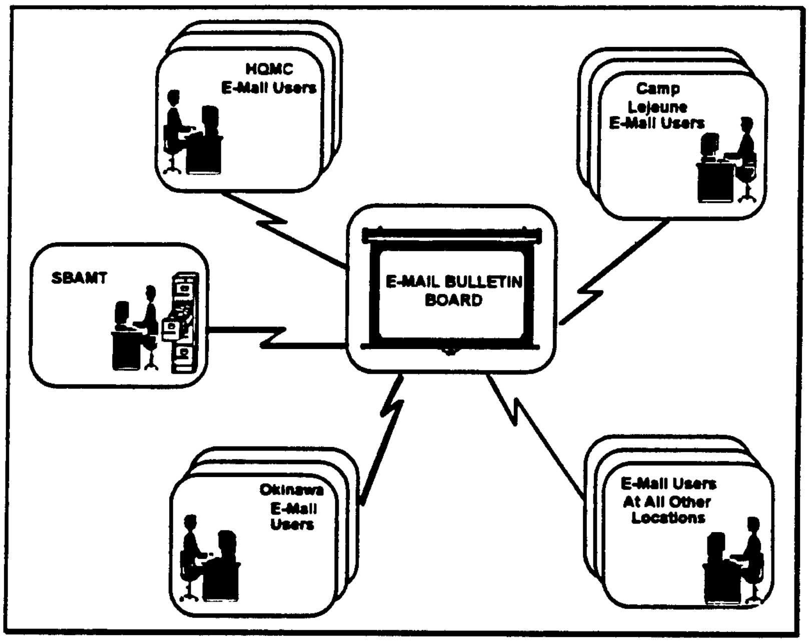


Figure H-4. The E-mail Bulletin Board Posts All SBA News for All Personnel to Access

EIS applications

The development of an SBA Executive Information System (EIS) is another effective communication tool. The primary focus of such a system is to provide an electronic means of keeping senior management aware of changes in SBA projects.

The typical EIS system is easy to use, has user-defined triggers and a myriad of other features that make such a system a very useful tool. (For example, each executive can define areas of particular interest so that when one of his SBA projects is affected in any way, an electronic message is sent to his computer; similarly, other changes that are not of interest never show up on his screen).

Architecture remodeling

When should you remodel? When any of the principles developed in the architecture framework phase have changed. Another reason could be a major change in technology significant enough not to have been anticipated in the target architecture phase; however, such changes will become increasingly rare. One of the major benefits of standards planning is that standards, unlike the underlying technology itself, change far more slowly.

In theory, one should never have to change the architecture if the architecture principles do not change; however, they do change from time to time. When this happens, the SBAMT should discuss and confirm the perceived changes with the SBA executive sponsor and all IT project managers before taking any action.

This page intentionally left blank.

Appendix I: Sample Deliverable Table of Contents

This section provides general outlines for each of the deliverables in the SBA planning process. These may be amended and customized by the AWG for presentation to the ASC. The individual circumstances surrounding the organizational culture and IT environment will also influence the deliverable.

The standards-based architecture

The standards-based architecture is composed of seven deliverables, which are released on a phased basis. Figure I-1 outlines the individual components of the model.

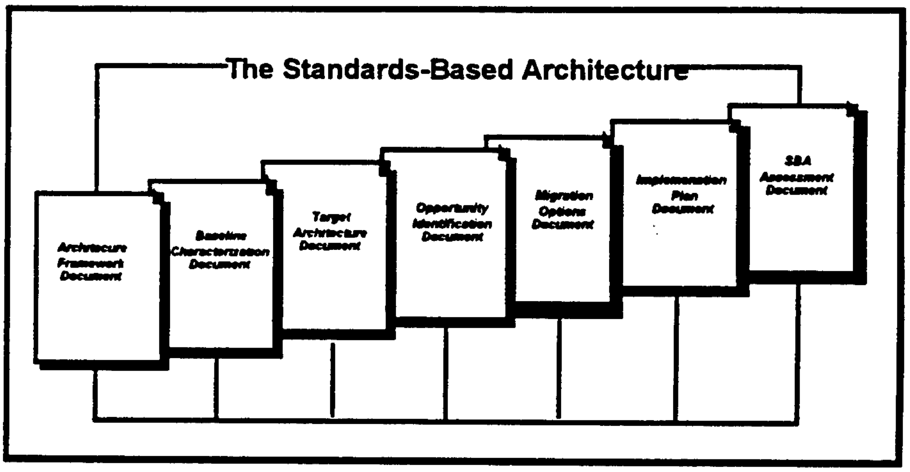

The Standards-Based Architecture
Architecure Framework Document
Baseline Characterization Document
Target Architecture Document
Opportunity Identification Document
Migration Options Document
Implementation Plan Document
SBA Assessment Document

Figure I-1. The Standards-Based Deliverable Set

Staged deliverables throughout the process

A key aspect of the standards-based planning process is the manner in which the architecture is developed. It is recommended that at each phase of the planning process an interim deliverable be produced by the team. Figure I-2 illustrates the phases and their associated deliverables.

Deliverable style

All of the deliverables should be "executive style" in scope, easy to read, and highly visual in nature. The key attribute of these deliverables is that they are distributed across the organization and are used to communicate the chief attributes of the architecture to the various constituencies within the enterprise.

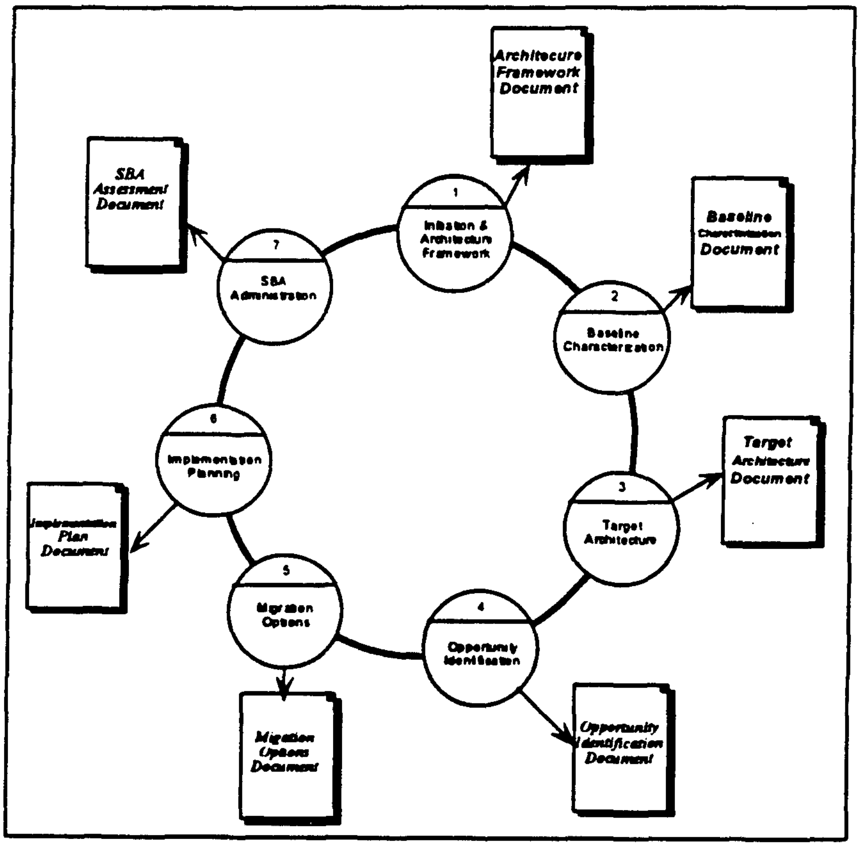

Architecure Framework Document
SBA Assessment Document
1
Initiation & Architecture Framework
Baseline Characterization Document
7
SBA Administration
2
Baseline Characterization
Target Architecture Document
6
Implementation Planning
3
Target Architecture
Implementation Plan Document
5
Migration Options
4
Opportunity Identification
Migration Options Document
Opportunity Identification Document

Figure I-2. The Standards-Based Deliverable Set

The length of each document should be between 25 and 45 pages. This will assure that the documents actually get read by individuals in the organizations.

Architecture Framework Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Key functional drivers and issues

III. Key interview findings

IV. IT principles constitution

V. Architecture planning issues

- Functional technology issues
- IT description: current environment
- Security issues
- Cost/benefit design concerns

VI. Functional and information opportunities

VII. Design issues

- Design principles, guidelines, and implications
- Design alternatives review
- SBA design attributes

VIII. Next steps

Baseline Characterization Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Key architecture baseline characterization issues

III. Scope and approach

IV. Classification and description

- Platform classification
- Generic application model
- Generic technology model
- Work flow model
- Generic information model
- Standards support description
- Security evaluation
- Connectivity support model
- Cost/performance data

V. Summary assessment of design issues and constraints of current environment

VI. Implications for target architecture design

VII. Next steps

Target Architecture Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Target architecture description

- Work flow and processes
- Data and information
- Applications
- Technology platforms
- Standards
- Migration issues
- Architecture organization and personnel issues

III. Architecture design alternatives

IV. Procurement issues

V. Implementation issues

VI. Next steps

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Implementation opportunity identification

- Strategic opportunities
- Major opportunities
- Quick hits
- General benefit and business case
- Magnitude, payoff, and degrees of freedom classification

III. Overall benefit classification

IV. Next steps

Migration Options Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. General cost/benefit definition

III. Migration project scope definition

IV. Technology standard implementation strategy

V. Time lines and trigger points

VI. Project cost and time frame considerations

VII. Specific business case and cost/benefit analysis for identified opportunities

VIII. Project deliverables definition

IX. Organizational change process requirements

X. Next steps

Implementation Plan Document(s)

This is not a formal presentation document, rather it is the aggregate set of project plan documents produced by the individual functional unit.

Presented below is a suggested set of topic areas to include in each plan. These may vary widely depending upon the implementation project but should comply with all DoD project management standards.

I. Project description

II. Objectives

III. Scope

IV. Deliverables

V. Critical success factors

VI. Constraints

VII. Task list

VIII. Effectiveness measures

IX. Technology requirements

X. Staffing skills

XI. Completion criteria

XII. Other issues

SBA Assessment Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Scope of architecture review

III. Key review findings

IV. Implementation adherence to IT principles and target architecture

- Processes
- Information
- Platforms
- Standards
- Migration issues
- Architecture organization and personnel issues

V. User views of benefits and functionality delivered

VI. Review of cost/benefit implementation delivered

VII. Continuous process improvement recommendations

VIII. Next steps

This page intentionally left blank.

Appendix J: Glossary

American National Standards Institute (ANSI): The principal standards coordination body in the United States. ANSI is a member of the International Organization for Standardization (ISO).

Application: The use of capabilities (services and facilities) provided by an information system specific to the satisfaction of a set of user requirements. [P1003.0/D15]

Application Entity: The part of an application process that interacts with another application process.

Application Layer: Layer seven of the OSI Reference Model. It serves as a window through which applications access communication services.

Application Model: A term used to describe those functions of an organization that can be supported or automated through IT. It is used for grouping or clustering functions into applications. It provides the application developers' views of the IT architecture.

Application Process: The part of an application that resides in a single end system.

Architecture: Architecture has various meanings depending upon its contextual usage.
(1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE STD 610.12]
(2) Organizational structure of a system or component. [IEEE STD 610.12]
(3)The disciplined definition of the IT infrastructure required by a business to attain its objectives and achieve a business vision. It is the structure given to information, applications, and organizational and technological means—the groupings of components, their interrelationships, the principles and guidelines governing their design, and their evolution over time.

Bridge: The hardware and software used to connect circuits and equipment in two networks with the same protocol.

Common Applications Environment (CAE): The X/Open term for a computer environment in which applications can be ported across X/Open vendor systems. It includes standards for the operating system, languages, networking protocols, and data management.

Computer-Aided Acquisition and Logistics Support (CALS): Standards for electronic file format interchange and data management adopted by the U.S. Department of Defense to acquire, process, and disseminate technical information in digital form. CALS will facilitate the transfer of logistic and technical information between industry and Government by leveraging existing international standards. Among the industry standards used in CALS are IGES (CAD, vector graphics), SGML (automated publishing), GRP 4 Raster or TRIF (raster scanned images), and CGM (illustrations).

Computer-Aided Software Engineering (CASE): A set of software tools that automate and contribute to the improvement of the software development process.

Conformance: Meeting standards. By running standard test scripts, conformance testing ensures that a product meets standards.

Connection: In data communications terminology, a logical link established between application processes that enables them to exchange information. In the OSI Reference Model, an association established by one layer with two or more entities of the next higher layer for the transfer of data. In TCP/IP, it is a logical TCP communication path identified by a pair of sockets, one for each side of the path.

Data Link: An assembly of two or more terminal installations and an interconnecting line.

Data Link Layer: Layer two of the OSI Reference Model. It controls the transfer of information between nodes over the physical layer.

Directory Services: A service of the External Environment entity of the Technical Reference Model that provides locator services that are restricted to finding the location of a service, location of data, or translation of a common name into a network specific address. It is analogous to telephone books and supports distributed directory implementations. [TA]

Distributed System: A system consisting of a group of connected, cooperating computers.

Distribution List: A list containing the names of mail users and/or other distribution lists. It is used to send the same message to multiple mail users. It can be private or public.

Electronic Mail: The electronic generation, transmission, and display of correspondence and documents. Electronic mail is a GAE.

Entity: An active element within an open system layer (e.g., session entity, transport entity). It can represent one layer, one part of a layer, or several layers of the OSI Reference Model. One layer can include several entities.

Exterior Gateway Protocol (EGP): The service by which gateways exchange information about what systems they can reach.

Gateway: A device for converting one network's message protocol to the format used by another network's protocol. It can be implemented in hardware or software.

Generic Application Environment (GAE): A term used to describe the set of architecture components that describe the different possible types of IT applications.

Generic Technology Environment (GTE): A term used to describe the set of architecture components that describe the different types of services required to support a GAE.

Generic Technology Platform (GTP): A term used to describe the different types of delivery components that can be used to support IT applications.

Government Open Systems Interconnection Profile (GOSIP): A government (e.g., U.S. or U.K.) profile of functional applications that outlines a national policy and strategy for converting to a communications system based on OSI. Use of GOSIP is no longer mandatory.

Host: A computer, particularly a source or destination of messages, on a communications network.

Information Model: A term used to describe the information resources of the organization and their interrela-tionships. It is used to support data modeling and resulting database and document storage design requirements. It provides the information resource managers' views of the architecture.

Institute of Electrical and Electronics Engineers (IEEE): An accredited standards body that has produced standards such as the network-oriented 802 protocols and POSIX. Members represent an international cross section of users, vendors, and engineering professionals.

Integrated Services Digital Network (ISDN): The recommendation published by CCITT for private or public digital telephone networks where binary data, such as graphics and digitized voice, travel over the same lines. ISDN will unite voice and data transmission, including imaging, over the same kind of digital network that links most telephone transmissions in use today.

Interface: A connecting link between two systems. In the OSI Reference Model, it is the boundary between adjacent layers.

International Standard (IS): Agreed international standard as voted by ISO.

International Organization for Standardization (ISO): An organization that establishes international standards for computer network architecture. Its OSI Reference Model divides network functions into seven layers. (Membership is by country, with more than 90 countries currently participating.)

Interoperability: (1) The ability of two or more systems or components to exchange and use information. [IEEE STD 610.12]. (2) The ability of the systems, units, or forces to provide and receive services from other systems, units, or forces, and to use the services so interchanged to enable them to operate effectively together. The conditions achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. [Joint Pub 1-02, DoD/NATO] [JOPES ROC]

(2)The ability of applications and computers from different vendors and architectures to work together on a network.

Interoperability Testing: Procedures for ensuring that a computer product or system can communicate in a multivendor network.

Layer: A level of the OSI Reference Model. The model divides functions for transferring information between systems into seven layers, grouping the related functions or tasks and making them easier to understand. Each layer performs certain tasks to move the information from sender to receiver. Protocols within the layers define the tasks for networks but not how the software accomplishes the tasks. Interfaces pass information between the layers they connect.

Local Area Network (LAN): A data network, located on a user's premises, within a limited geographic region. Communication within a local area network is not subject to external regulation; however, communication across the network boundary may be subject to some form of regulation. [FIPS PUB 11-3]

Message: A block of information sent from a source to one or more destinations.

MS-DOS: The personal computer operating system developed by Microsoft Corporation.

Multivendor Network: A computer network with hardware and software from more than one vendor.

National Institute for Standards and Technology (NIST): The division of the U.S. Department of Commerce that ensures standardization within Government agencies. NIST is responsible for the Applications Portability Profile—a set of standards and guidelines for U.S. Government procurement. NIST was formerly known as the National Bureau of Standards (NBS).

Network: A system of connected computers.

Network Layer: The third layer of the OSI Reference Model. This layer controls underlying telecommunication functions such as routing, relaying, and data link connections.

Node: A point in a network, either at the end of a communication line (end node) or where two lines meet (intermediate node).

Open Network: A network that can communicate with any system component (peripherals, computers, or other networks) implemented to the international standard (without special protocol conversions, such as gateways).

Open Software Foundation (OSF): An organization created by major IT vendors to define specifications, develop software, and make available an open, portable environment.

Open Systems: (1) A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [P1003.0/D15] (2) Software environments consisting of products and technologies that are designed and implemented in accordance with "standards" (established and de facto) that are vendor independent and commonly available.

Open Systems Interconnection (OSI): A set of standards that, when implemented, let different computer systems communicate with each other.

Operating System: A group of programs operating under the control of a data processing monitor program. It manages such functions as memory, processing tasks, and interprocess communication in a computer system.

OSI Reference Model: The seven-layer model, defined by the ISO, that provides the framework for building an open network. The seven layers, ranging from highest to lowest, are application, presentation, session, transport, network, data link, and physical.

Password: A string of characters required to gain access to directories, files, or applications.

Peer Protocol: The protocol governing communications between program entities that have the same function in the same layer in each of two OSI networks.

Physical Layer: The first layer of the OSI Reference Model. It governs hardware connectors and byte-stream encoding for transmission. It is the only layer that involves a physical transfer of information between network nodes.

Portable Operating System Interface for Computer Environments (POSIX): An IEEE standard operating-system interface defining the external characteristics and facilities required to achieve the portability of applications at the source-code level.

Portability: (1) The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE STD 610.12] (2) A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment. [IEEE TUTOR] (3) The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TA]

Porting: The process by which a software application is made operational on a computer architecture different from the one on which it was originally created.

Presentation Layer: The sixth layer of the OSI Reference Model. It allows an application to properly interpret the data being transferred.

Process: A general term for any computer operation on data.

Profile: A set of one or more base standards, and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [P1003.0/D15]

Protocol: A set of rules governing network functionality. The OSI Reference Model uses sets of communication protocols to facilitate communication between computer networks and their components.

Quality of Service (QOS): A set of characteristics of a connection as observed between the connection end points. In the OSI session and transport layers, acceptable QOS values are negotiated between the service users when the connection is established.

Scalability: The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII] The capability to grow to accommodate increased work loads.

Server Type: A class of servers in a client/server architecture.

Service Provider: The resource that provides the facilities of the relevant OSI Reference Model layer. The OSI session and transport layers are the service providers for the session and transport services, and the X.25 network gateway or X.25 message control system is the service provider for the network service.

Service User: The software application using the facilities of one of the layers of the OSI Reference Model. For example, a program that calls the programmatic interface to the session layer is a session service user.

Session Layer: The sixth layer of the OSI Reference Model. It provides the means for two session service users to organize and synchronize their dialogues and manage the exchange of data.

Store-and-Forward Message System: The communication process that allows messages to be stored at intermediate nodes before being forwarded to their destination. X.400 defines a message handling system that uses this process.

System:–People, machines, and methods organized to accomplish a set of specific functions. [FIPS PUB 11-3]

TCP/IP Gateway: A device, or pair of devices, that interconnects two or more networks or subnetworks, enabling the passage of data from one (sub)network to another. In this architecture, a gateway contains an IP module and, for each connected subnetwork, a subnetwork protocol (SNP) module. The routing protocol is used to coordinate with other gateways. A gateway is often called an IP router.

Technology Model: A term used to define and describe the components of the technology infrastructure that support the other architecture models. It is in this area that the enabling effect of standards-based architectures is felt the most. The technology model provides the technology managers' views of the architecture.

UniForum: A trade association dedicated to promoting UNIX and open systems. UniForum sponsors UNIX events, publishes magazines, directories and technical overviews, and proposes specifications.

UNIX: An operating system that has become a de facto industry standard, supported on a wide range of hardware systems from a variety of vendors.

UNIX International: The consortium that defines and promotes the UNIX operating system and related software products.

Wide-Area Network (WAN): A public or private computer network serving a wide geographic area.

Work Organization Model: A term used to describe the impact on business operations at the work group and user

levels. It is used by organizational change designers to manage the impact of introducing new IT systems. It provides the users' views of the architecture.

X.25: Recommendations developed by CCITT that define a protocol for communication between packet-switched public data networks and user devices in the packet-switched mode.

X.400: The international standard for a store-and-forward message handling system in a multivendor environment.

X/Open Company Ltd.: *A* nonprofit corporation made up of vendors and large corporate users who are investing in the specification of the X/Open Portability Guide (XPG), an open environment based on standards. X/Open also brands products.

This page intentionally left blank.

Appendix K: Proposing Changes to TAFIM Volumes

Introduction

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown below. Guidance on the use of the format is subsequently provided.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown below, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail:
TAFIM
Booz•Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/671-7937**; indicate "TAFIM" on cover sheet.

TAFIM Change Proposal Submission Format

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

Format Guidance

The format should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**a**) of the format would identify that individual. The information in the **Document Identification** part of the format (**b**) is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts (**c**, **d**, or **n**) of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: 'The working baseline will only be available to the TAFIM project staff.'" The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The c (5), d (5), or n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in c (6), d (6), or n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

Application Function – Template

Instructions:

List all applications identified on the Application Inventory template in the left-most column of this template (give the common *application abbreviation* and *full name* if known).

Major functions as identified in MHSS Technical Management Plan document are listed in the column headings.

If there are other functions that your area performs which are not covered by these columns, list them in the blank columns.

Place an "X" at the intersection of the row and column if the application in that row provides some automated information systems support to the business function represented by the column.

Function *Existing Application*	Health Care Delivery Clinical	Health Care Delivery Preventative	Health Care Management	Financial	Logistics	Human Resources	Information	Research	Education	Liaison		

Application to Function – Sample

Function / Existing Application	Command & Control	Doctrine	Finance	Human Services	Intelligence	Logistics	Mainten-ance	Manpower	Plans/ Concepts	Policy	Procure-ment	Requirements	Supply	Training/ Education	Transport	Warfare/ Operations/ Combat Simulation
ABA			X													
ABA				X												
ABE						X										
ACJS																
AFRS																
ALPS																
AMHS					X											
AMMOLOGS																
AN/TPS-59 RADAR											X		X			
AOWP																X
APADE						X										
APCS							X									
ARMS									X							
ATAC•																
ATRIMS																
AV-3M/NAVFLIRS						X		X				X		X		
AWIS						X										
AWN																
BAS																
BCS																
BCS																X
BNA								X	X	X						X
BNA								X	X	X						
BNA					X			X	X	X						
BNA								X	X	X						
BREES						X										
BUDGET																
BUDS CLASS II																
CAEMS									X				X		X	
CAEMS									X				X		X	
CAIMS																
CAIS				X												
CASPRO								X								
CATS										X						
CCS													X			
CDCS											X		X			
CMIS									X				X			
CPMS				X												
DAIS				X												
DASC																
DCERPS																
DCIP			X													
DDA						X										
DE													X			
DEERS IIB				X												
DEERS IIIB				X												
DEPMEDS													X			

Application to Physical Location – Template

Instructions:

List all applications identified on the Application Inventory template in the left-most column of this template (give common application abbreviation and full name if known).

List all known physical locations where these applications may be accessed across the top of the matrix, one per column.

If more columns are needed repeat the applications on a additional sheet(s) and continue listing the locations on those sheets.

Place an "X" at the intersection of the row and column if the application in that row is accessed by at least one user at the location represented by the column.

NOTE: Access to the application at a location does not necessarily mean the application is running on a computer at that location. It could be remote.

Physical Location *Existing Application*										

Application to Physical Location - Sample

Physical Location / Existing Application	1st MAW	29 Palms	2nd MA	3rd MAW	4th MAW	Albany	BN Level	Cherry Point	El Toro	Parris Island	Pendleton	Quantico
ABA												
ABA											x	
ABE								x	x			
ACIS												x
AFRS												x
AIMS												x
ALPS						x					x	x
AMHS												
AMMOLOGS						x				x	x	
AN/TPS-59	x		x	x	x							
AOWP												x
APADE								x	x			
APCS						x						
ARMS											x	
ATAC+												
ATRIMS											x	x
AV-3M/NAVFLIRS								x	x		x	
AWIS												
AWN												
BCS		x										
BCS												
BNA												x
BNA												x
BNA												
BNA												x
BREES								x	x			
BUDGET						x					x	x
BUDGET CLASS II						x					x	x
CAEMS												
CAIMS												
CAIS												x
CASPRO												
CATS								x	x			
CCS						x					x	x
CDCS												
CMIS											x	
CPMS						x						
DAIS												x
DASC	x		x	x	x							
DCIP												
DDA								x	x			
DE						x					x	x
DEERS IIB						x					x	x
DEERS IIIB						x					x	x
DEPMEDS						x						

Application to Standard – Template

Instructions:

For all applications identified on the Application Inventory Template fill in the standards which are in use currently.

User Interface The software (or hardware) which "presents" the results of the application system's processing to the user (i.e. MS-Windows, 3270, Character Based, etc.) Include any online monitor software such as CICS or IMS/DC in this column as well.

Database The data base or file management software used by the application (i.e. Oracle, DB2, dBase IV, Adabas, VSAM, ISAM, etc.)

Application Any self-contained "package" software which has been made an integral part of the application (i.e. Commander EIS, MS-Excel, Harvard Graphics, etc.)

Language What the application code is written in (i.e. Cobol, Cobol II, C, C++, Pascal, ADA, etc.)

Operating System The system software under which the application runs (i.e. MS-DOS, MVS-XA, DOS/VSE, OS/400, OS/2, UNIX System V, Xenix, etc.)

Communications Media The transmission path for data communications (i.e. Marine Corps Data Network, Defense Data Network , Autodin, Token Ring, Arcnet, LocalTalk, etc.)

Communications Protocol The protocol(s) used to control data communications related to this application (i.e. Banyan Vines, Novell Netware, SNP, SDLC, SNA, LU6.2, etc.)

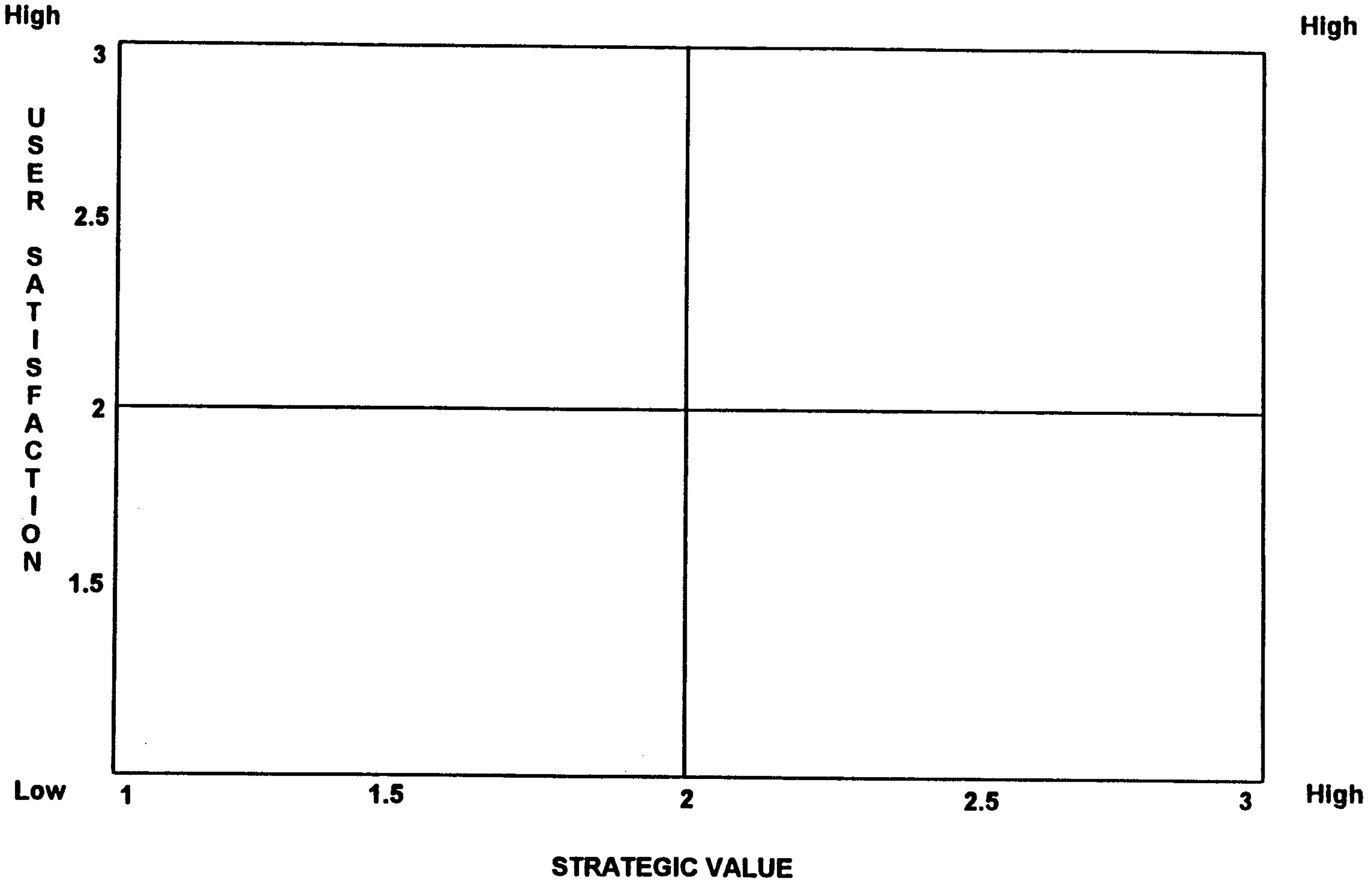
Other Services Any other aspects of the application which do not fall under any standards listed in prior columns

Standard / *Existing Application*	User Interface	Database	Application	Language	Operating System	Commun. Media	Commun. Protocol	Other Services

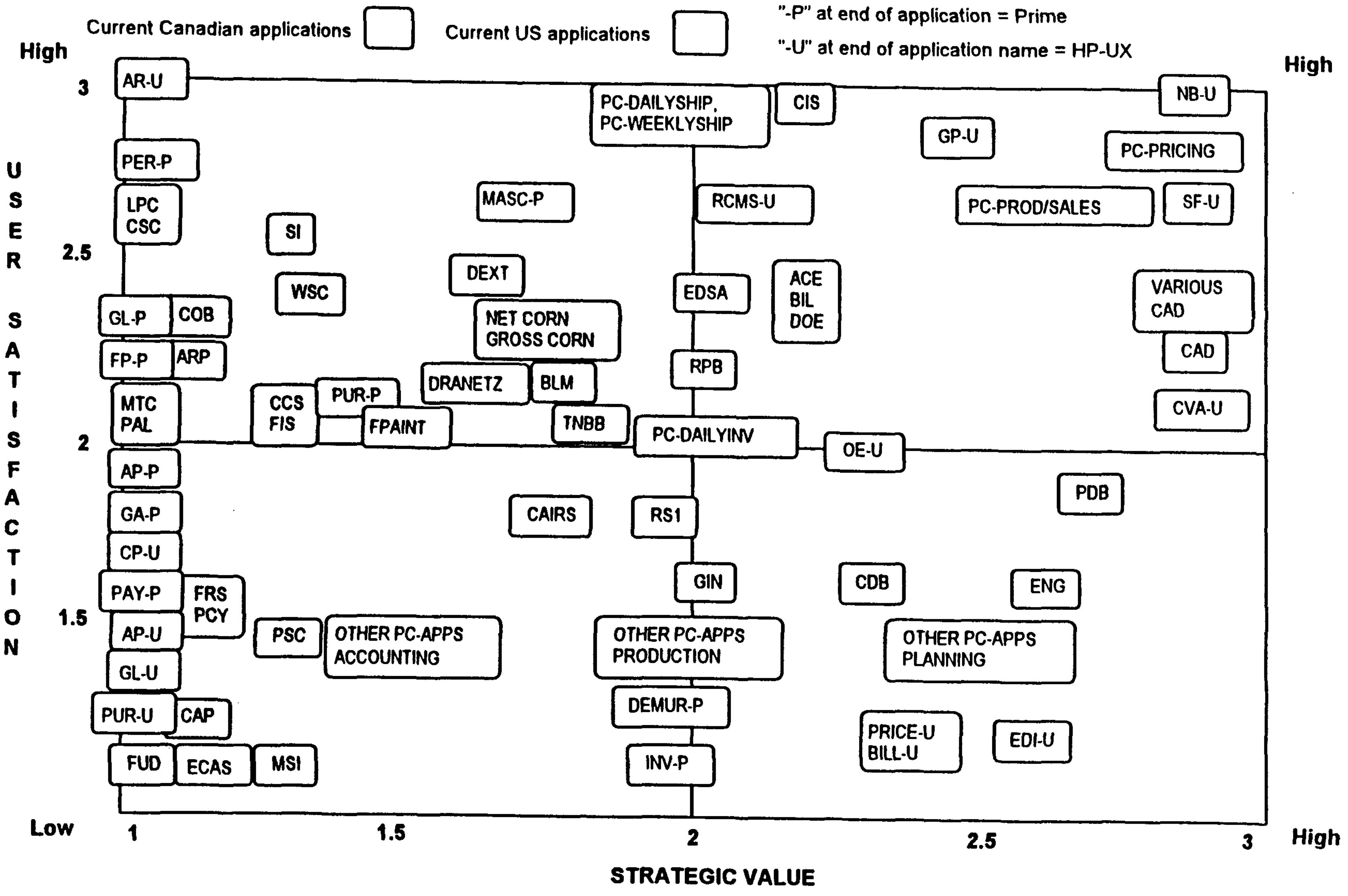
Application to Standard - Sample

Standard / Existing Application	User Interface	Database	Application	Language	Operating System	Commun. Media	Commun. Protocol	Other Services
UADPS-WCS	UTS-30 (Sperry Terminals)	DMS 1100		Cobol	EXEC (Univac)	NAVNET	SDLC	
CATS	DPS (Display Processing	DMS 1100		Mapper	EXEC (Univac)		SDLC	
DDA	PC / 6530 (Tandem			TACL	Guardian		SDLC / 3270	
ABE	TDI			TACL	Guardian		Async / 6530	
G-MAN	PC / 3270			TACL	Guardian			
MDAS	PC / 6530 (Tandem			TACL	Guardian	NLN (Navy Logisitics Net)	SDLC	
RODS				TACL	Guardian			
UADPS-SP	PC / 3270			Cobol	MCP (Burroughs)			
MCCRES	CICS	DBaseIII		ADA	MS-DOS	MCDN		
SWIFTHAWK				C	MS-DOS			
SORTS	3270	Adabase		Cobol	MS-DOS			
MAGTF II				Cobol	MS-DOS	WWMCCS		
SASSY IA				Cobol	MS-DOS			
MCLLS		DBaseIII		DBase / Clippe	MS-DOS	MCDN		
TCAC				Fortran	MS-DOS	VAF/UHF		
EPOS	3270 Emulation			PC Focus	MS-DOS		SNA SYS	Bar Code
OCIS				PC Focus	MS-DOS			
RPM/FHS		Oracle		SQL	MS-DOS			
MTF EDITOR				Turbo Pascal	MS-DOS			
MIDAS (EIS)			Redimaster		MS-DOS		Async	
CMIS			Harvard Graphics		MS-DOS			
AV-3M/NAVFLIRS	CICS			ADA / Cobol II	MS-DOS / MVS XA		TDI	
ATRIMS	CICS	Adabase		Ada	MVS-XA	MCDN		
MCAIMS	CICS	Adabase		C / Adabase	MVS-XA	MCDN		
AMMOLOGS	CICS/Natural	Adabase		Cobol	MVS-XA	MCDN		
MEDLOGS	CICS/Natural	Adabase		Cobol	MVS-XA	MCDN / Autodin		
MIMMS IB	CICS/Natural	Adabase		Cobol	MVS-XA	MCDN		
JUMPS/MMS	CICS / Natural	Adabase		Cobol	MVS-XA	MCDN	SNA LU6.2	
BUDGET				Cobol II	MVS-XA	Tape / Disk Transfer		
ATAC +					MVS-XA	Token Ring	VINES	
TMS	CICS/Complete/ Natural	Adabase		Cobol / Natural	MVS-XA / MS-DOS	MCDN		
APADE	MCR (Borroughs Master Control program)			TACL	Pathway	Token Ring	VINES	
PLRS				CMS2Y	SDEX / M	UHF	PLRS Interface Controller	
PLRS	PLRS Unique	PLRS Unique	Navigator CMD & Control	CMS-2	SDEX/M, SDEX	PLRS Unique	PLRS Unique	
NALISS	CICS			Natural			SDLC / 3270	
BCS				TACFIRE S/W		Comm. Wire	TACFIRE S/W	

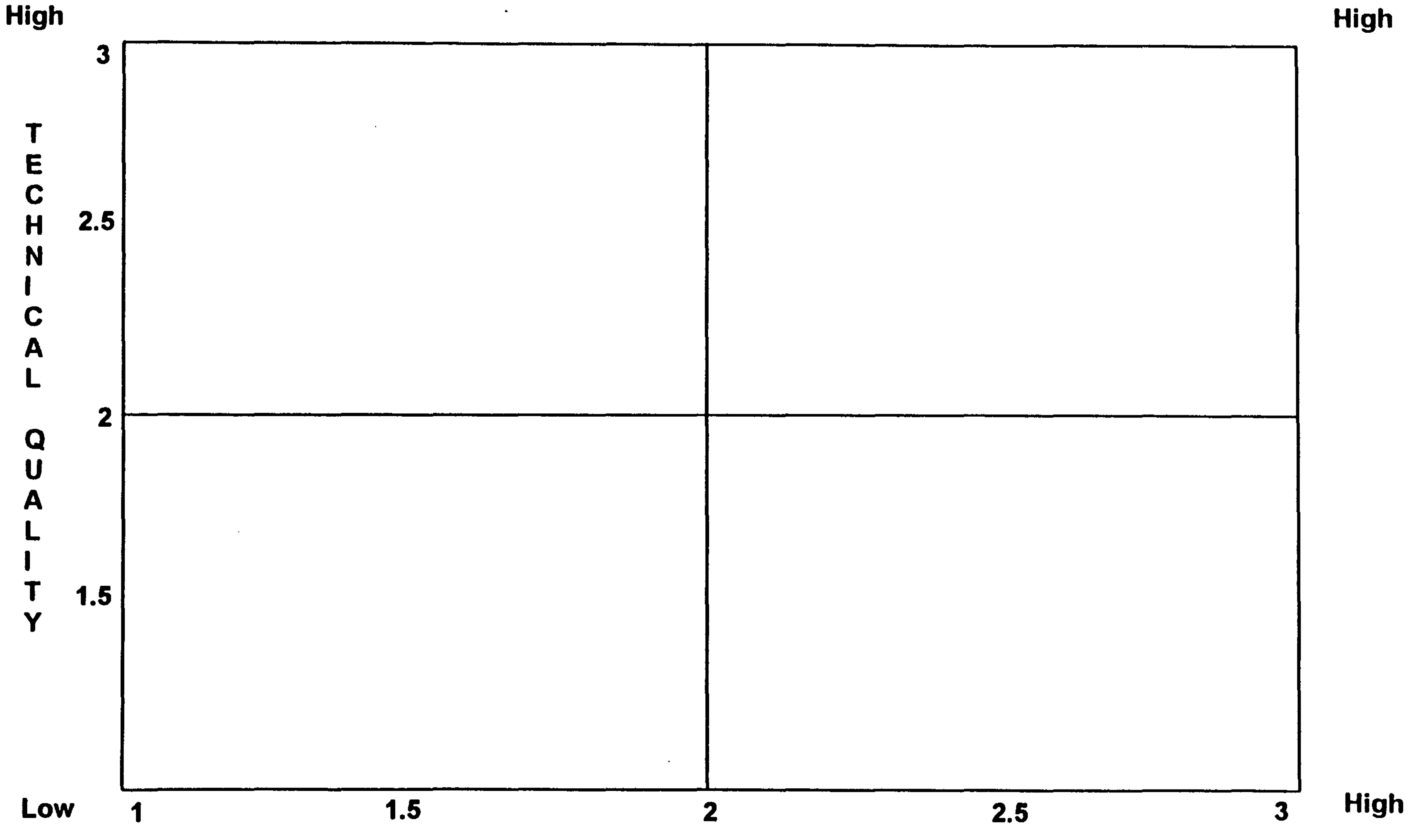
User Satisfaction versus Strategic Value



User Satisfaction versus Strategic Value

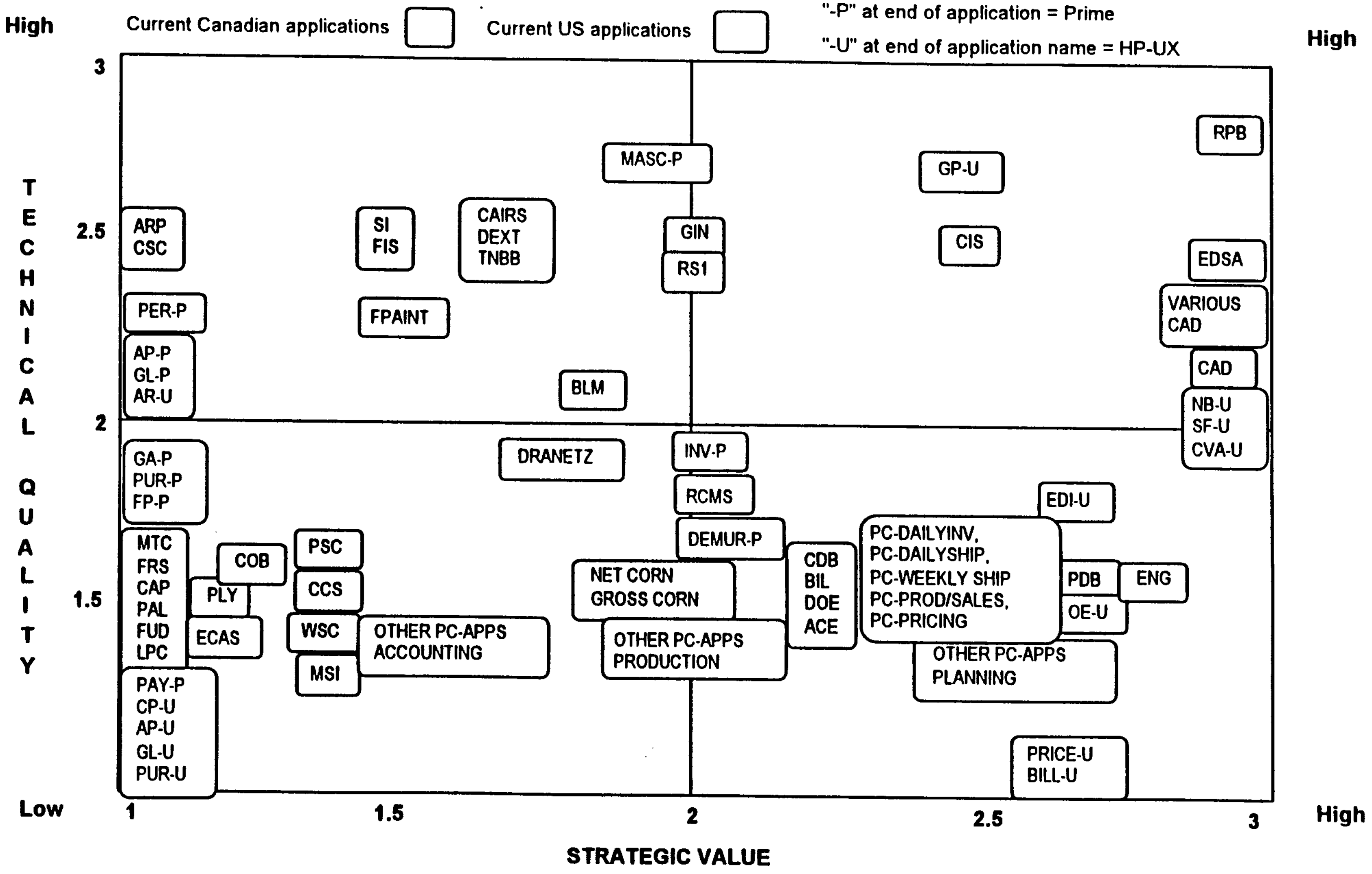


Technical Quality versus Strategic Value

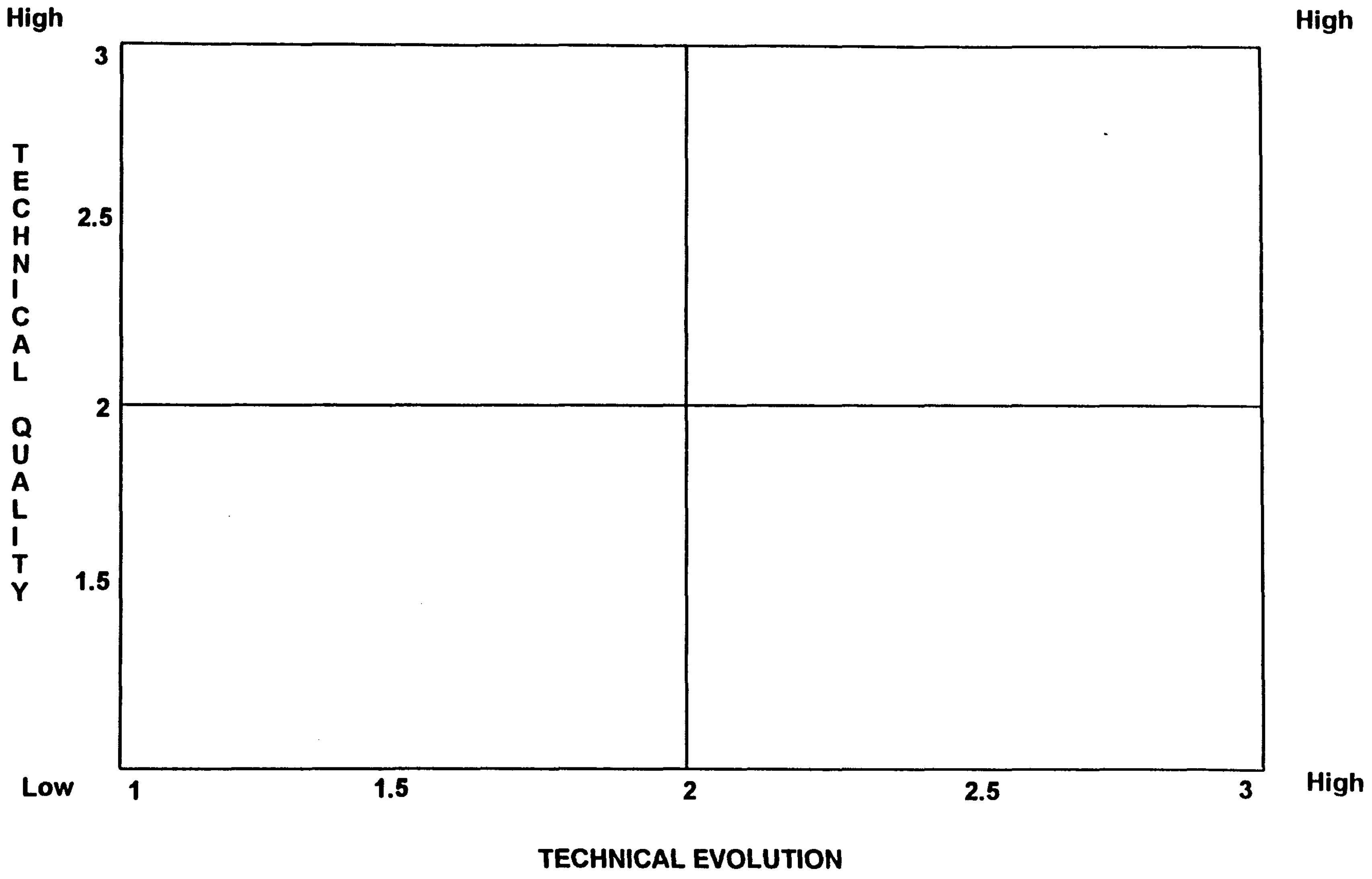


STRATEGIC VALUE

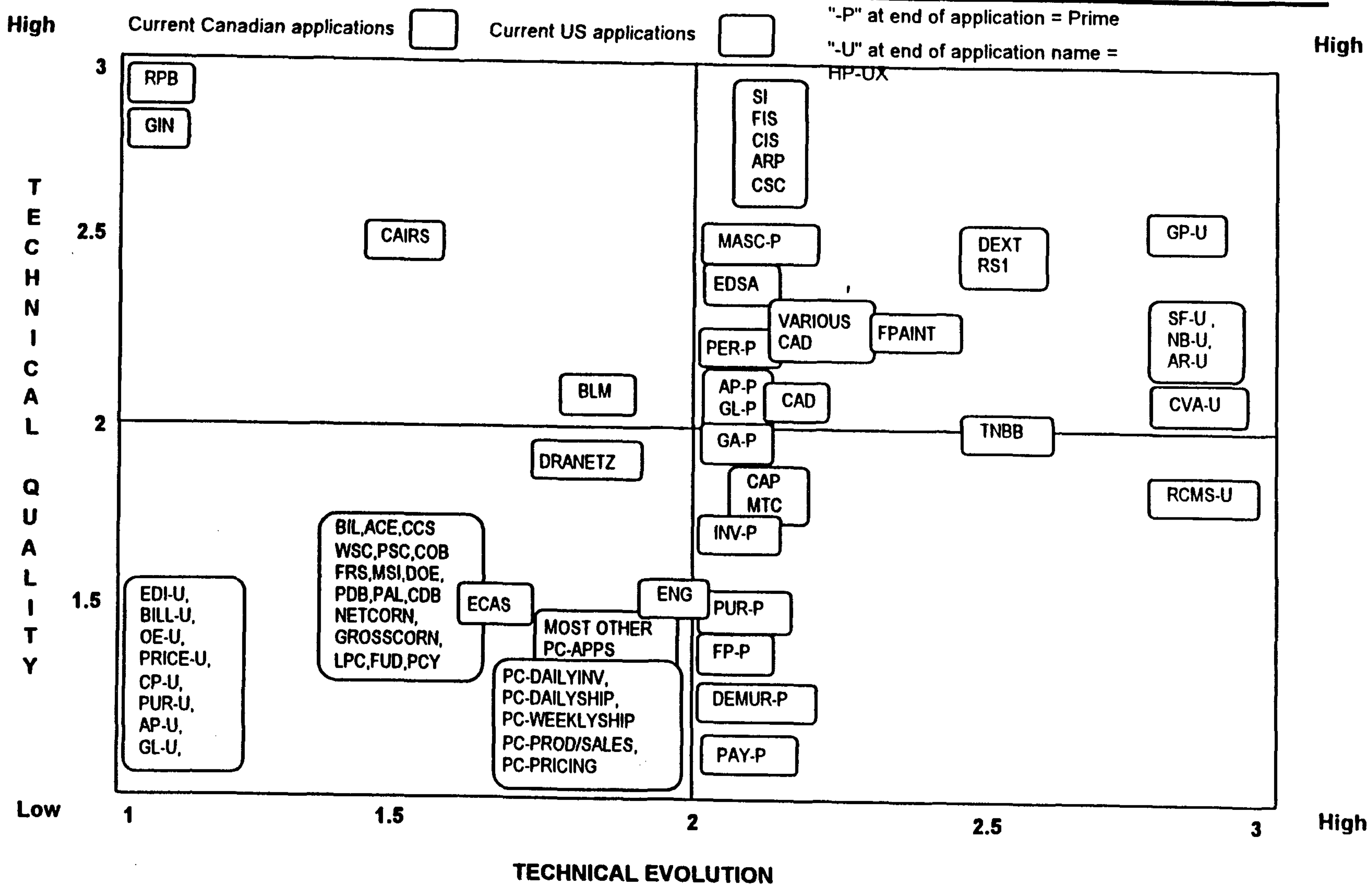
Technical Quality versus Strategic Value



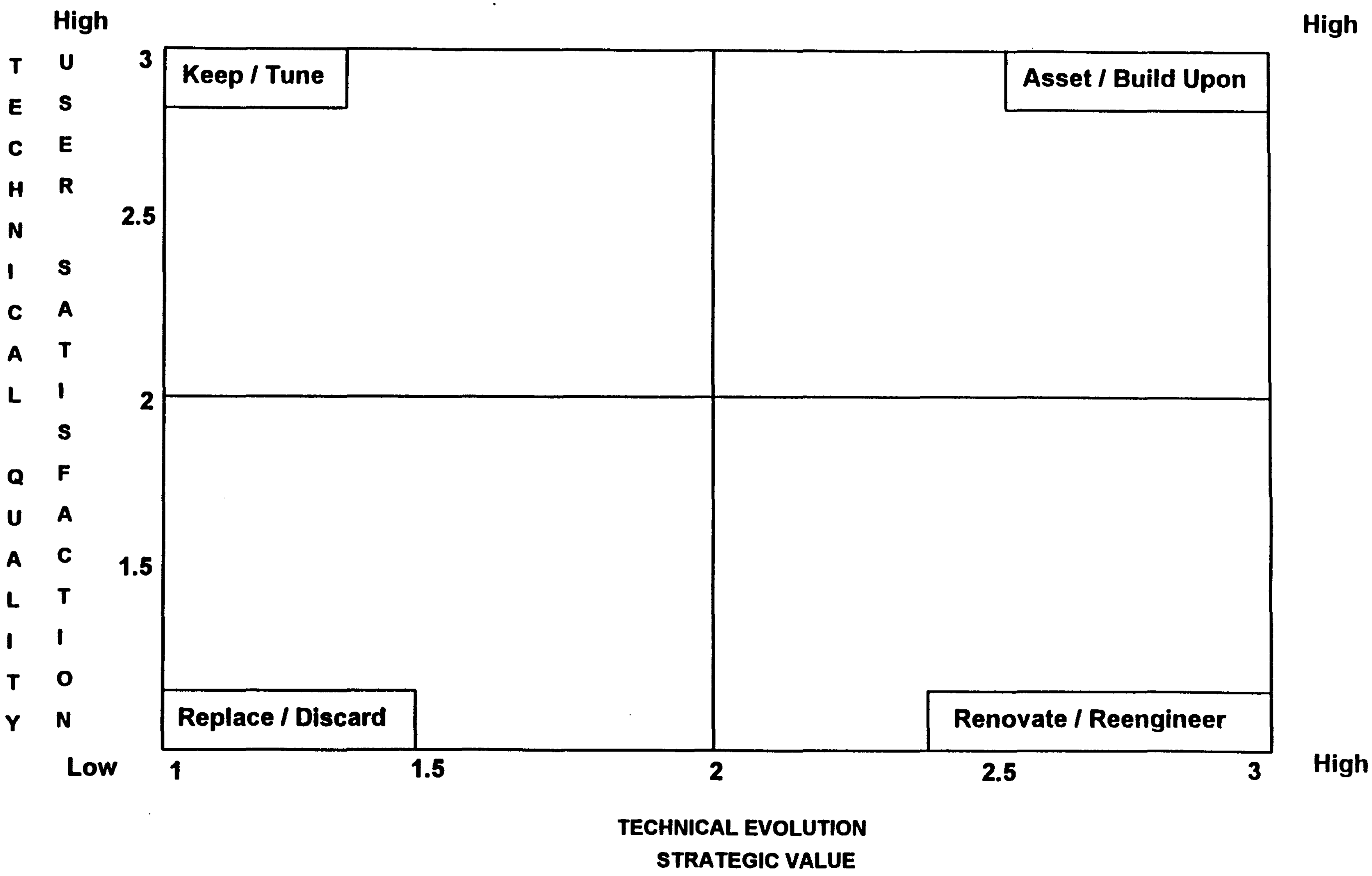
Technical Quality versus Technical Evolution



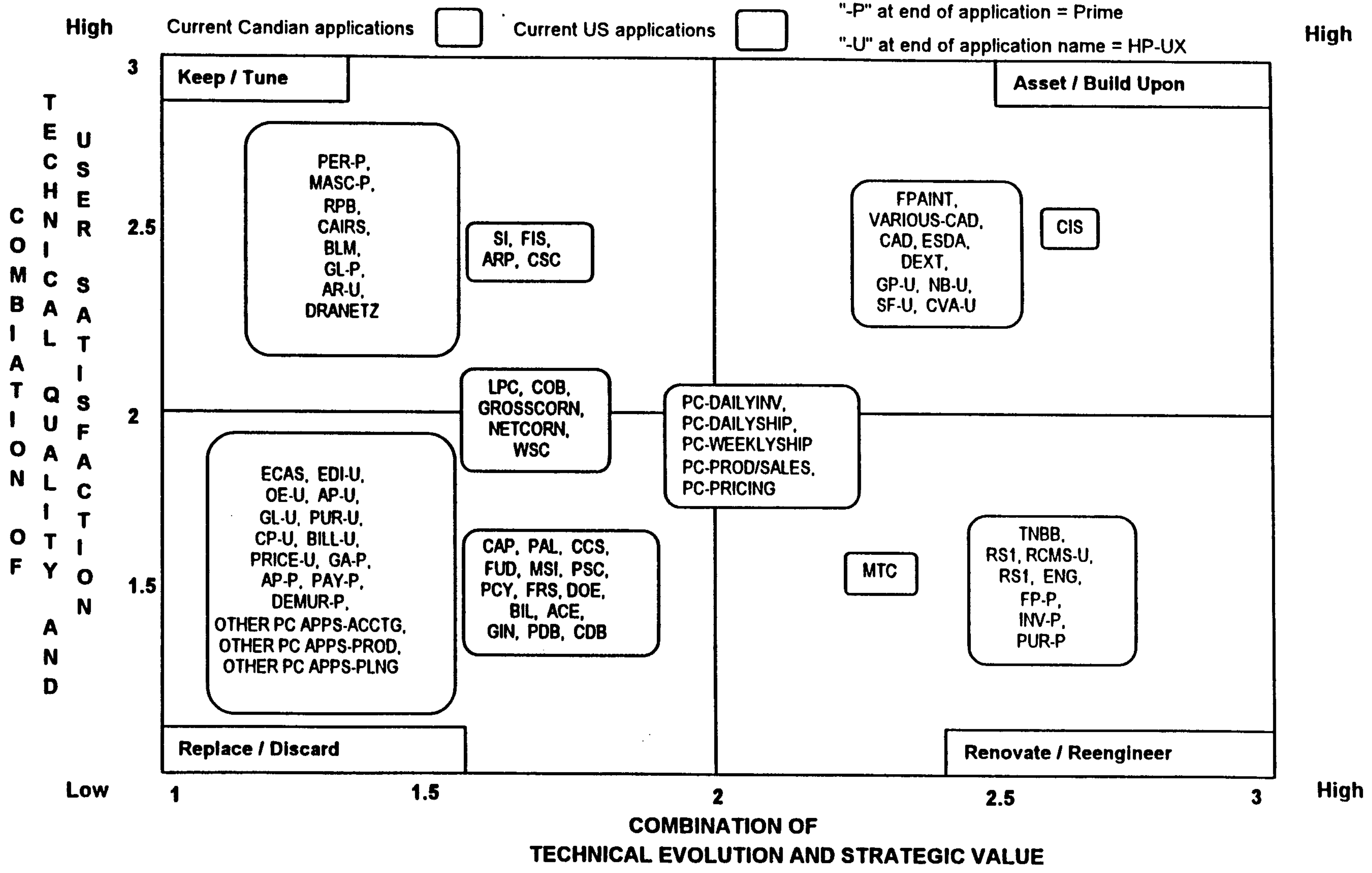
Technical Quality versus Technical Evolution



Summarized Application Assessment



Summarized Application Assessment



Current Candian applications □ Current US applications □

"-P" at end of application = Prime

"-U" at end of application name = HP-UX

Keep / Tune

PER-P,
MASC-P,
RPB,
CAIRS,
BLM,
GL-P,
AR-U,
DRANETZ

SI, FIS,
ARP, CSC

LPC, COB,
GROSSCORN,
NETCORN,
WSC

Asset / Build Upon

FPAINT,
VARIOUS-CAD,
CAD, ESDA,
DEXT,
GP-U, NB-U,
SF-U, CVA-U

CIS

PC-DAILYINV,
PC-DAILYSHIP,
PC-WEEKLYSHIP
PC-PROD/SALES,
PC-PRICING

Replace / Discard

ECAS, EDI-U,
OE-U, AP-U,
GL-U, PUR-U,
CP-U, BILL-U,
PRICE-U, GA-P,
AP-P, PAY-P,
DEMUR-P,
OTHER PC APPS-ACCTG,
OTHER PC APPS-PROD,
OTHER PC APPS-PLNG

CAP, PAL, CCS,
FUD, MSI, PSC,
PCY, FRS, DOE,
BIL, ACE,
GIN, PDB, CDB

Renovate / Reengineer

MTC

TNBB,
RS1, RCMS-U,
RS1, ENG,
FP-P,
INV-P,
PUR-P

High
Low
COMBINATION OF TECHNICAL QUALITY AND USER SATISFACTION
3
2.5
2
1.5
1
1.5
2
2.5
3
High
COMBINATION OF
TECHNICAL EVOLUTION AND STRATEGIC VALUE

Technology Templates

This page intentionally left blank.

Personal Computer Workstation Inventory – Template

Instructions:

Fill in the column headings with all known physical locations where Personal Computers and/or Workstations are in use.

Place a quantity at the intersection of the row and column to depict the number of items in use at the physical location represented by the column.

The following definition of rows apply:

Number of Users The number of users who use or potentially could use the PCs and/or workstations as this location

IBM Compatibles The number of IBM PCs or Compatibles by CPU type (i.e. XT,286,386,486,586) in use at this location

Other The number of any other type of PC or Workstation (one row for each specific Vendor/Model) in use at this location

Total PC's and/or Workstations The sum of the preceding rows, providing the total number of PCs and/or Workstations in use at this location

Number Connected to LANs The number of PCs and/or Workstations which are connected to a Local Area Network at this location

Number of LANs The number of discrete Local Area Networks in use at this location, without regard to whether these LANs are interconnected

Number of LANs Connected to WANs The number of Local Area Networks at this location which have connectivity to other remote locations via a Wide Area Network

PC and/or Workstation Owner The Person and/or Organization with the budgetary ownership or responsibility for the PC's and/or Workstations at this location

PC and/or Workstation Manager The Person and/or Organization with the day-to-day operations responsibility for the PC's and/or Workstations at this location

	Physical Locations											
Inventory Item:												
Number of Users												
IBM Compatibles												
XT												
286												
386												
486												
586												
Others (list Vendor/Models below)												
Total PCs and/or Workstations												
Number Connected to LANs												
Number of LANs												
Number of LANs Connected to WANs												
PC and/or Workstation Owner												
PC and/or Workstation Manager												

Personal Computer and Workstation - Sample

Inventory Item	Physical Locations									
	29 Palms MCAGCC	Camp Lejeune 2nd Div	Camp Lejeune 2nd FSSG	Camp Lejeune CG,MCB	Camp Lejeune II MEF	Camp Pendleton MCTSSA	Camp Pendleton 1st Div	Camp Pendleton 1st FSSG	Camp Pendleton CG,MCB	Camp Pendleton I MEF
Number of Users										
IBM Compatibles										
XT	16	11	40	128		20	95	71	31	14
286	116	626	608	967	216	344	872	718	722	159
386	89	157	118	223	33	204	211	322	238	80
486	8	17	61		4	59	18	70		19
586										
Others (List Vendor/Models Below)										
Apple Macintosh	2			4		10			2	1
Total PCs and/or Workstations	231	811	827	1322	253	637	1196	1181	993	273
Number Connected to LANs	173	608	620	991	189	477	897	885	744	204
Number of LANs	22	17	20	32	12	10	33	52	70	25
Number of LANs Connected to WANs	22	14	16	32	8	10	29	44	70	20
PC and/or Workstation Owner	RJE	G6/ISMO	G6/ISMO	RASC	G6/ISMO	MARCOR	G6/ISMO	G6/ISMO	RASC	G6/ISMO
PC and/or Workstation Manager	RJE	G6/ISMO	G6/ISMO	RASC	G6/ISMO	MARCOR	G6/ISMO	G6/ISMO	RASC	G6/ISMO

Standards to Platform – Template

Instructions:

Describe the specific standards in place for your Information Technology Platforms in the categories depicted by the Generic Technology Platform names in each row.
Fill in the column headings with the same Platform names used on the Technology Platform Inventory Template. This should be a unique identifier for the specific technology platform.
At the intersection of the Generic Technology Platform and the Platform Name enter the specific standard which is in place on that platform.
Where multiple platforms of the same type are used in a similar fashion, list the Platform Name once and the approximate number of units in parentheses (i.e. IBM-PC486 (150 units)).
The following row definitions apply:

User Interface The standard(s) which controls the presentation of the results of computer system processing to the user (i.e. MS-Windows, 3270, Character-based, etc.
Operating System The standard(s) which controls the basic operation of the computing platform (i.e. MS-DOS, MVS-XA, DOS/VSE, OS/400, OS/2, UNIX System V, Xenix, etc.)
Communications Management The standard(s) which controls the connectivity of this platform to others (i.e. Banyan Vines, Novell Netware, SNA, TCP/IP, SMP, SDLC, etc.)
Data Base (and/or file) Management The standard(s) which controls how the data is managed on the platform (i.e. DOS File System, dBase, Adabas, Oracle, Paradox, DB2, etc.)
Transaction Monitor The standard(s) which controls the processing of online transactions (i.e. CICS, IMS-DC, TSO, CMS, etc.)
Document Management The standard(s) which controls document creation, storage and retrieval for the platform (i.e. DISOSS, MS-Word, Word Perfect, Office Vision, etc.)
Distribution Management The standard(s) which controls the distribution of user messages and/or files within this platform and to other interconnected platforms (i.e. Vines Mail, Quickmail, Office Vision, DISOSS, etc.)
Conferencing Management The standard(s) which controls the resources of the platform to allow computer conferencing, shared screen conferencing, etc.
Development Services The standard(s) which controls the environment under which computer programs are developed and tested (i.e. compilers, toolkits, CASE tools, debugging aids, etc.)
Repository Services The standard(s) which controls the metadata, and data which describes the overall information management environment (i.e. IBM's MVS Repository Manager, IMS-Data Dictionary, etc.)
Notes Any additional notes which will help clarify the specific platform standards in place or planned for the near term

	Platform Names					
Standard						
User Interface						
Operating System						
Communications Management						
Data Base (and/or file) Management						
Transaction Monitor						
Document Management						
Distribution Management						
Conferencing Management						
Development Services						
Repository Services						

Notes: ______________________________
____
____
____
____
____
____

Standards to Platform – Sample

	Platform Names					
Standard	286, 386, 486 Micro/PCs	Network Servers (LAN)	AS/400 (mini)	Hewlett Packard 3000 (mini)	Mid-Size Mainframe	Large Mainframe
User Interface	Various Proprietary S/W	Various Proprietary S/W	Proprietary S/W	Proprietary S/W	TSO	TSO/ROSCOE/ CICS
Operating System	MS-DOS	Banyan Vines	OS/400	MPEV - VDelta9	MVS/SP	MVS-XA
Communications Management	Enable/Pro Comm	Vines NOS	IBM 3270 Emulation	NS-3000		VTAM
Data Base (and/or file) Manageme	Enable/dBase IV		SQL Compliant	Turbo Image 3000 (now SQL)		ADABASE
Transaction Monitor						CICS
Document Management	Enable		Office Vision			DISOSS
Distribution Management		Vines Mail Program	Office Vision			DISOSS
Conferencing Management						
Development Services	ADA/Clipper	Vines ToolKit	RPG/Cobol	HP-TRANSACT/Cobol		ADA/Cobol/ Assembler
Repository Services				Dictionary 3000		ADABASE

Notes: Within 6 months all "Mid-Sized Mainframe" systems will be replaced by "IDNX Boxes" & channel extended off the Marine Corps Large Mainframe systems.

Exceptions: RJE Iwakuni JA

RJE Kaneohe HI

RJE CCDH, HQMC Washington DC

Technology Inventory - Template

Instructions:

Generic Technology Platforms are listed in the leftmost column as row identifiers. *There are likely to be more than one specific instance of each of these Generic Technology Platforms,* i.e. within workstation, we may have Various kinds of terminals (NOTE: PC's and other Intelligent Workstations should not be included on this template. They are covered on the Personal Computer and Workstation Inventory Template). We may have multiple Vendors (i.e. IBM, Teletype Corp, Hitachi, HP, etc.). Within each vendor there could be multiple models (i.e. IBM 3278, IBM 3279, TLX 178-2, etc.)

And each of these unique non-intelligent workstations would have an "owner", i.e. the person or organization which has budgetary ownership or responsibility for these workstations.

Enter one line for each unique combination of Platform Name, Vendor Name, Platform Model, and Platform Owner. Use additional pages if you exhaust the blank rows within a Generic Technology Platform section.

Enter each physical location which can contain one or more of these technology platforms as column headings under the broad heading "Quantities by Physical Location".

Enter the quantity of each specific Platform (as depicted by a unique combination of Platform Name, Vendor Name, Platform Model, and Platform Owner) in the intersection with the column depicting the physical location which houses the platform.

Use additional pages if more columns are needed to depict physical locations.

The Following definition of the Generic Technology Platforms apply:

Workstation Any terminal device which is not a PC or Intelligent Workstation (i.e. "dumb" or single-purpose terminals which have not been included on the Personal Computer and Workstation Template)

Output/Input Peripheral Platforms such as Laser Printers, Line Printers, Scanners, Card Readers, etc.

*Local Area Network (LAN)*The hardware aspects of Local Area Networks such as Token Ring, Ethernet, LocalTalk, etc.

LAN Server A platform which is used to run the LAN operating system and services. This could be a PC or a specialized server platform of any size.

*Wide Area Network (WAN)*A device which provides long haul communications services, such as DISN, DDN, Telenet, Tymnet, etc.

Network Interface Device A device which provide an interface between the network and the computer processor, such as Front-end processors like IBM 3705 and NCR Comten.

*Concentrator, Multiplexer/*Devices such as routers, gateways, cluster controllers, etc., which allow the basic network resources to be *effectively used for multiple purposes and devices at a various points in time.*
Switching Device

Transmission Device Specialized devices which provide the transmission medium for information transfer, beyond the facilities identified earlier in the WAN category, such as VHF and SHF Radio and/or Satellite.

Storage Device Devices which allow the storage and retrieval of information, such as Disk Drives, *Tape Drives, Microfilm Processors, etc.*

Mid-Range Processor Processors which are larger than workstation or terminal devices both physically and in processing and peripheral controlling capability, but not of the large, mainframe class, i.e. AS/400, 4381, HP9000, etc.

Large Processor The largest mainframe processing platforms such as IBM's 3090 Model 400E and Hitachi's 5890-600E. This would also include large massively parallel processors and/or vector based supercomputers.

Image Processor Devices which are specifically devoted to the processing of digital images, such as Kodak's Komstar system.

Identifying Information					Quantities by Physical Location												
Technology Platform	**Platform Name**	**Vendor Name**	**Platform Model**	**Platform Owner**													
Workstation																	
Output/Input Peripheral																	
Local Area Network																	
LAN Server																	
Wide Area Network																	
Network Interface																	
Concentrator/ Multiplexer/ Switching																	
Transmission Device																	
Storage Devices																	
Mid-Range Processor																	
Large Processor																	
Image Processor																	

Technology Inventory – Sample

Identifying Information					Quantities by Physical Location										
Technology Platform	Platform Name	Vendor Name	Platform Model	Platform Owner	29 Palms MCAGCC	Albany IRMD	Barstow	Blount Island	Camp Foster	Camp Lejeune FASC	Camp Lejeune RASC	Camp Pendleton FASC	Camp Pendleton RASC	Cherry Point RASC	El Toro RASC
Workstation	Terminal	HDS	3472-FC1	C4I, HQMC											
Workstation	Terminal	HP	2392-A	C4I, HQMC											
Output/Input Peripheral	Laser Printer	IBM	3812-2	C4I, HQMC											
Output/Input Peripheral	Laser Printer	Xerox	9700	C4I, HQMC							2				
Local Area Network (LAN)	10 Base-T	10 Base-T		RJE 29 Palms	22										
Local Area Network (LAN)	Ethernet	3Com		1st. FSSG								*N/A			
LAN Server	Banyan FS	IBM	386/486	1st. FSSG								*N/A			
LAN Server	Banyan FS	IBM	386/486	2nd. FSSG						*N/A					
Wide Area Network (WAN)	MCDN	USMC		MCCTA, Quantico	1	1	1	1	1	1	1	1	1	1	1
Network Interface Device	FEP	Comten	NCR 3690-DS	C4I, HQMC			1								
Network Interface Device	FEP	Comten	NCR5675	C4I, HQMC		1					1		1		
Concentrator/ Multiplexer/ Switching Device	Digital Switch (Tactical)	Tactical	AN/TTC-42	MEF											
Concentrator/ Multiplexer/ Switching Device	Digital Switch (Tactical)	Tactical	SB-3865	MEF											
Transmission Device	HF MUX- Radio Central (semi-mobile)	Tactical	AN/TSC-120 (proposed)	MEF											
Transmission Device	VHF-Single Channel-Manpack Radio	Tactical	AN/PRC-68	MEF											
Storage Devices	DASD	Amdahl	62/6880	C4I, HQMC		7									
Storage Devices	DASD	HP	3937	C4I, HQMC											
Mid-Range Processor	Midrange	HDS	EX33	C4I, HQMC											
Mid-Range Processor	Midrange	HP	3000-58	C4I, HQMC											
Large Processor	Mainframe	Amdahl	5890-300E	C4I, HQMC											
Large Processor	Mainframe	Amdahl	5890-600E	C4I, HQMC		1							1		
Image System	Komstar	Kodak	Model IV	C4I, HQMC		1					2		1		
Image System	Komstar	Kodak	Model VI	C4I, HQMC											

Cost Templates

This page intentionally left blank.

Application System Costs – Template

Instructions:

Fill in the column headings with the application acronyms which are represented on the Application Inventory Template. Use additional pages as needed.

The leftmost column provides the cost categories for which we need actual costs or headcount related to each application system.

At the intersection of a cost category and application system enter the costs (or headcount if the category relates to FTEs).

If you share any of these categories with other organizations, allocate your proportionate share of support and place a note at the bottom of the template, explaining the rationale for your allocation.

For example, if an outside contractor supplies a Help Desk with 25 Full Time Equivalents (FTEs) but you only use 4 of the total FTEs, record the 4 FTEs in the Support Service Contractors FTE Headcount.

The following definitions apply to the cost categories:

Hardware **Processors, terminals, PCs, Workstations, Frontend Processors, Disk/Tape Drives, etc.**

Software **Online Monitors, DBMS's, Compilers, Report Writers, Operating Systems, etc.**

Applications **Commercial packages and custom-developed application systems which provide end-user functionality**

Maintenance (Hardware) **Maintenance costs on all of the above**

Internal Direct Support **Staff costs directly associated with developing, maintaining and operating the above items**

Support Service Contractors **Supplemental staff beyond those listed for in-house staff shown above**

Network **Includes owned/leased equipment, Public Switched Network, VAN's and other network services**

Internal FTE Headcount **Internal full time equivalent headcount**

Support Service Contractors (FTE Headcount) **Full time equivalent headcount for supplemental staff beyond internal staff identified above**

Other ***Any other item not covered in above categories (please explain with a note)***

	Applications				
Cost Categories					
Hardware					
Software					
Applications					
Maintenance (Hardware)					
Internal Direct Support					
Support Service Contractors (Cost)					
Network					
Internal FTE Headcount					
Support Service Contractors (FTE Headcount)					
Other					
Total Costs					

Notes: ______________________________

____

____

____

____

____

Application System Costs – Sample

	Applications				
Cost Categories	*SASSY*	*APCS*	*MCFMIS*	*RPM/FHS*	*TCAIMS*
Hardware			1,340K	4,450K	17K
Software		40K	100K	1,800K	
Applications					
Maintenance (Hardware)				170K	
Internal Direct Support	1,650K		760K	70K	1,640K
Support Service Contractors (Cost)	2,400K	40K		1,260K	2,760K
Network					
Internal FTE Headcount					
Support Service Contractors (FTE Headcount)					
Other					
Total Costs	4,050K	80K	2,200K	7,750K	4,417K

Notes: ______________________________

Security Templates

This page intentionally left blank.

Application to Security Level:

- *Enter all Applications as identified on the Baseline Template for Application Inventory*
- *Selected Security Classifications are across the top. If these need modification, please mark the headings approprpriately.*
- *If other security classifications should be tracked, use blank column.*
- *Enter an "x" in intersection of platform type and column to denote the presence of security at this level for this application.*
- *If more columns are needed for additional security classifications or rows for additional applications, use a second sheet.*
- *If there are specific exceptions or notes for clarification, enter the information in the space below the matrix.*

This template should apply to all applications in the entire enterprise.

Security Classification *Application*	Top Secret	Secret	Confidential	Unclassified Sensistive	Unclassified		

Additional Notes and Clarifications:

Location to Security Level:

- *Enter all Physical Locations as identified on the Baseline Template for Business Functions and Work Organization*
- *Selected Security Classifications are across the top. If these need modification, please mark the headings approprpriately.*
- *If other security classifications should be tracked, use blank column.*
- *Enter an "x" in intersection of platform type and column to denote the presence of security at this level for this work location.*
- *If more columns are needed for additional security classifications or rows for additional locations, use a second sheet.*
- *If there are specific exceptions or notes for clarification, enter the information in the space below the matrix.*

This template should apply to all locations in the entire enterprise.

Security Classification *Location*	Top Secret	Secret	Confidential	Unclassified Sensistive	Unclassified		

Additional Notes and Clarifications: ______________________________

____

____

____

____

____

Generic Technology Platforms to Security Level:

- *Locations are listed in the left column, one per row.*
- *Generic Technology Platforms are across the top*
- *If another security classification should be noted, add it to the key and to the appropriate column(s)*
- *Enter all applicable security classifications at the intersection of each row and column*
- *If more rows are needed for additional locations, use a second sheet.*
- *If there are specific exceptions or notes for clarification, enter the information in the space below the matrix.*

This template should apply to all platforms in the entire enterprise.

Technology Platforms *Locations*	Workstation	Output/Input Device	Local Area Network (LAN)	LAN Server	Wide Area Network	Concentrator /Multiplexer /Switching	Storage Devices	Mid Range Processors	Large Processors

Additional Notes and Clarifications:

Security Classification Key:
TS = Top Secret
S = Secret
C = Confidential
US = Unclassified Sensitive
U = Unclassified

Baseline Template - Process to Security

Process to Security Level:

- *Enter all work processes as identified on the Baseline Template for Business Functions and Work Organization*
- *Selected Security Classifications are across the top. If these need modification, please mark the headings approprpriately.*
- *If other security classifications should be tracked, use blank column.*
- *Enter an "x" in intersection of platform type and column to denote the presence of security at this level for the activities and outputs of this work process.*
- *If more columns are needed for additional security classifications or rows for additional processes, use a second sheet.*
- *If there are specific exceptions or notes for clarification, enter the information in the space below the matrix.*

This template should apply to all processes in the entire enterprise.

Security Classification *Work Process*	Top Secret	Secret	Confidential	Unclassified Sensistive	Unclassified	Location	Misc.

Additional Notes and Clarifications: ______________________________

____

____

____

____

____

Questions of Interest and Rules of Thumb for Baseline Analysis

This page intentionally left blank.

Connectivity characteristics

- What is the relationship of the current platform to other target platforms? Is there a client/server relationship in place? If so, detail the associated platform environments and describe the client/server relationship.
- What are the characteristics of the logical links that the platform under review has to other platforms with which it is linked. Are the links based on peer-to-peer relationships such as LU 6.2? From a terminal interface perspective, how does the terminal view the platform linkage?
- What are the characteristics of the physical links that the platform under review has to other platforms with which it is linked? Are the links batch or interactive in nature? Interactive token ring or dial-up?
- What are the current characteristics of the existing platform with regard to the capacity of the platform under review—the number of transactions supported, effective throughput per period of time, bandwidth required, etc.?
- What are the problems and opportunities related to current connectivity attributes? Do they enable or inhibit platform performance? How do they relate to application, technology, or cost performance?

Standards support

What standards does the existing platform support and which of the following services?

- User interface services
- Database services
- Operating system services
- Communication services
- Management services
- Languages
- Applications.

- What interface standards does this platform support for all of the above (e.g. UNIX operating system support for POSIX P 1003.1 for operating system interface standard)?
- What effect does the current suite of standards have upon IT objectives? Are they enabling or hindering growth and attainment of IT objectives?
- What are the costs of using these standards? To what degree are our current standards "open"?
- To what degree are our standards vendor independent?
- What degrees of freedom do we have within our current standards as implemented in existing products or services?
- What is the nature of the standards supported? Are they proprietary or open?
- Are they de facto or de jure standards?
- How stable are they? Have they been in place for 6 to 24 months?
- Are they "developing" standards? If so, is the future standards path for this platform clear?
- To what degree does the existing platform either promote or inhibit portability, scalability, or interoperability?

Generic application environment support

- What generic application environments are currently supported on the existing platform?
- What are the logical linkages between existing GAEs?
- What are the problems and opportunities related to current application environment attributes—application delivery, technology, or cost-related issues?

Generic technology environment

- What are the existing generic technology components that make up the existing environment?

- What kinds of services are being supported by GTE?
- What classes of users are using which types of GTEs?
- What set of services is supported by the existing technology platform under baseline review?
- What are the problems and opportunities related to current generic technology attributes—application, technology, or cost-related? How mature are these environments? Where will these new applications go in the future?

GAE and GTE relationship: logical and physical connectivity

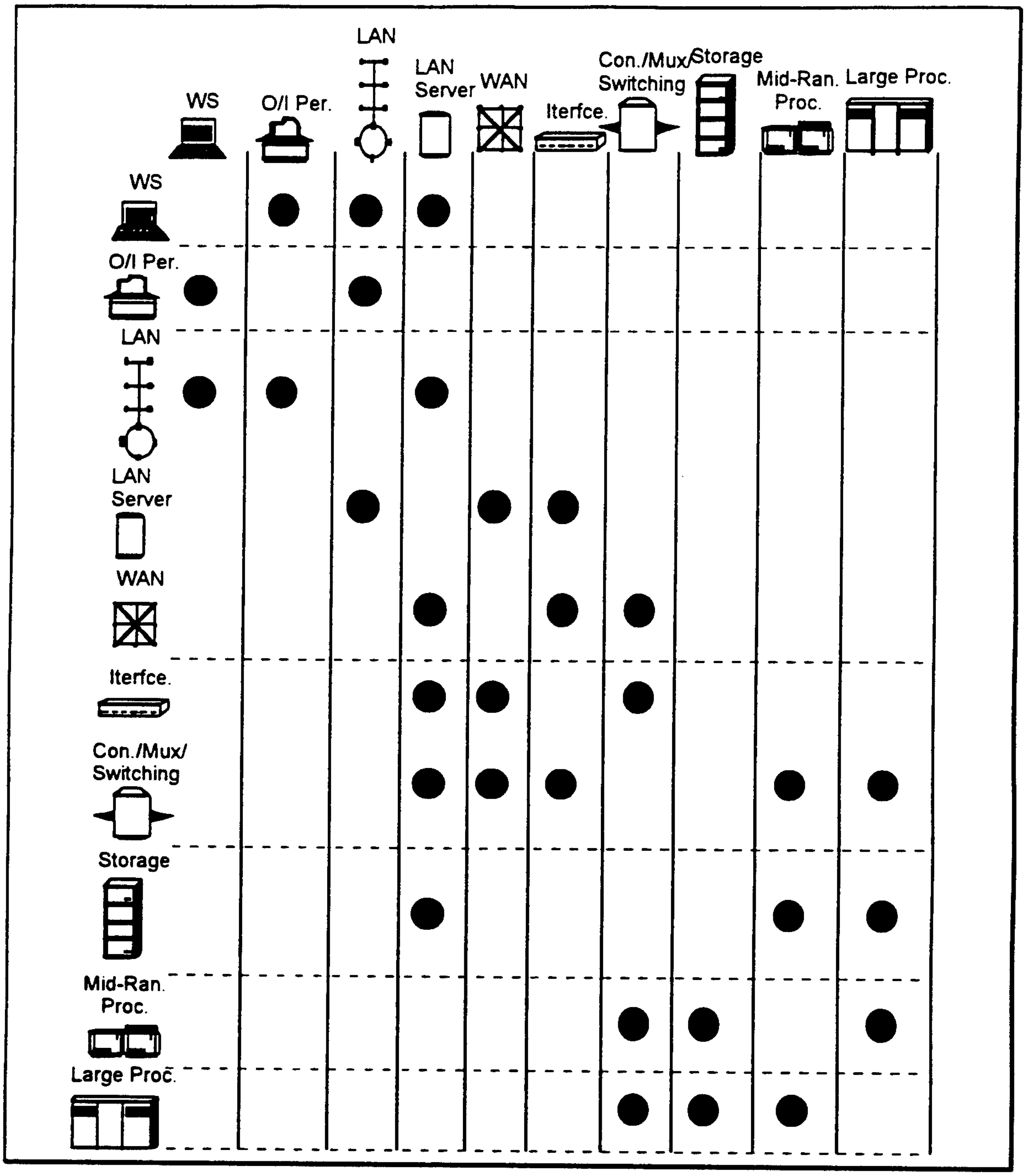
One of the key aspects of platform attributes is how individual platforms work together to "plug and play" in an overall architecture. In a function, every platform has a relationship with all other platforms in the function, even if they are standalone by nature. A method is needed to characterize these logical and physical relationships as well as their attendant costs in a simple visual manner. This is the essence of an architecture.

The problem in most functions is that the various platform attributes have not been decoupled from the technology itself and therefore do not lend themselves to a logical characterization for architecture planning and analysis. By examining each of the platform attributes on a logical as well as physical basis, we may develop an overall picture of how various platforms relate to one another across a function.

The following matrix demonstrates a typical three-tiered architecture in a hypothetical function composed of LAN-based work group computing, mid-range computers, and traditional mainframe access. Each one of the points on the matrix (dark dots) may be thought of as the logical connectivity point between the two platforms. By examining the individual attributes of each of the two platforms connected by these two points, one may examine the nature of platform connectivity. Indeed, using such a matrix, one may characterize a number of attributes in a baseline architecture:

- How GAEs are related between business units across a function.

- How GTEs are related between business units across a function.
- What standards are in place across a function or department.
- The physical or logical connectivity characteristics across a function as well as the relationship one processing environment has with another. For example, the "client/server" model may be illustrated in this manner.
- The relative cost of a platform or set of platforms in a function or department.



	WS	O/I Per.	LAN	LAN Server	WAN	Iterfce.	Con./Mux/ Switching	Storage	Mid-Ran. Proc.	Large Proc.
WS		●	●	●						
O/I Per.	●		●							
LAN	●	●		●						
LAN Server			●		●	●				
WAN				●		●	●			
Iterfce.				●	●		●			
Con./Mux/ Switching				●	●	●			●	●
Storage				●					●	●
Mid-Ran. Proc.							●	●		●
Large Proc.							●	●	●	

Figure B-1
Physical or Logical Connectivity

Platform cost data

- **Direct hardware costs**—purchase cost, depreciation or lease.
- **Direct operating system software costs**—purchase, cost, depreciation, or lease.

- **Maintenance and service costs**—recurring operational cases.
- **Personnel costs**—direct and indirect for both hardware and software.
- **Training costs**—direct and indirect for hardware and software.
- **Application costs**—direct and indirect including software licensing and maintenance as well as other "intangibles" related to work flow, business procedures, inventory turn rates, management "time value," and general productivity. (This last category will vary enormously depending upon how the application costs are quantified.)

Typically, cost data are the most difficult types of information to collect in an architecture assignment. There are many reasons for this including the fact that little useful cost information is kept in the first place. Good cost data can be very helpful, especially in justifying the migration plan later on. When cost data are available, they should be collected by the team to incorporate in the baseline characterization and for use in migration planning in later stages of the project. (see Appendix F for a full discussion of the business case cost analysis.)

Security evaluation criteria

A key aspect of the baseline includes providing a classification of the application and technology platforms using the criteria and classification scheme described in the *U.S. Department of Defense Trusted Computer System Evaluation Criteria* [DOD 5200.28 STD, December 1985]. The appendix includes an entire section of security planning considerations.

Rules of thumb for the baseline characterization

Once the matrices have all been completed, the analysis phase can begin in earnest. A substantial amount of effort must go into the analysis and interpretation of the baseline characterization to provide a solid platform for identifying the projects that will be required to move the enterprise from the baseline to the target architecture. The assessment of applications is of particular interest, since this provides the guidance that will be instrumental in improving the effectiveness of the enterprise. An example of application assessment is provided below in light of the significant role applications play in any architecture.

Baseline application assessment rules of thumb

To prioritize future application opportunities, an assessment of existing application systems is needed. In this way, the existing applications and their associated assessment can be mapped to the target application opportunities. For example, if an envisioned target application is of high strategic significance and the existing applications which provide equivalent functionality are assessed as being in need of replacement, the target application would be a high priority initiative in the migration and implementation planning phases. If there is no existing application and the other conditions described above for the target application were the same, the target initiative would be at an even higher priority.

This kind of analysis must be done for each target and existing application in the enterprise. To make this work, a high-level assessment of the existing applications is needed. The following provides some criteria that can be used for this process.

Templates should be provided to representatives of each functional area affected by the architecture and supporting IT staff, and should be completed with their assessment of any existing application systems with which they were familiar. Emphasis should be placed on doing them relatively quickly, with a reasonable subset of the user/developer community providing input. The intention is to perform the assessment only at a macro level for overall trends and conclusions. An example of such a template is shown below.

Categorizations

The recipients should be asked to categorize (high, medium, and low within each category) each application according to the following criteria:

- User satisfaction
- Technical quality
- Strategic value
- Technical evolution.

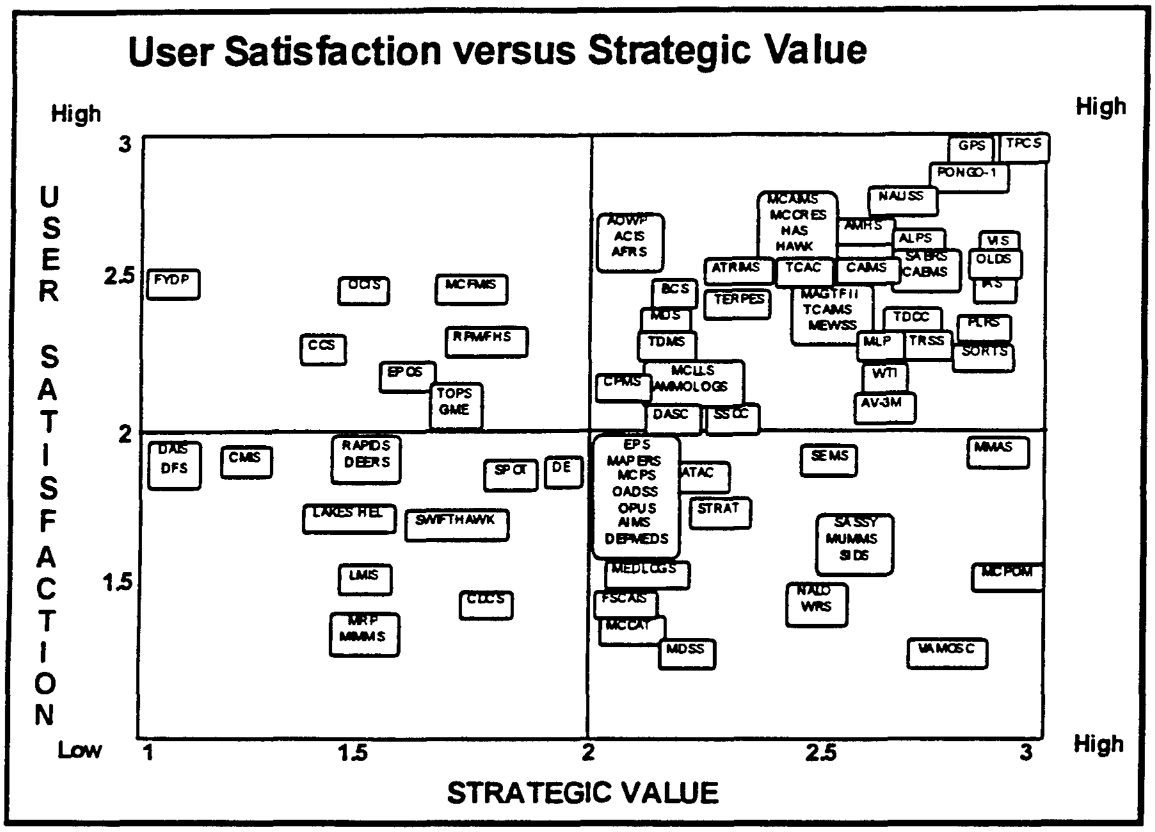


Figure B-2
Template for User Satisfaction Versus Strategic Value of Existing Applications

User satisfaction

User satisfaction is self explanatory. For the other categories, the following definitions should be provided:

Technical quality

This assessment criterion measures the application's robustness and maintainability. It is a measure of whether the application is well written with easy-to-follow, structured code and sufficient program comments to facilitate enhancements or maintenance. A high technical quality application will have data definitions (or other frequently changed items) included in the code as tables or copy members rather than hard coded within the programs. Similar logic will be coded once and referenced in other sections of the program or application rather than physically replicating. In general, a high level of technical quality would be an application that already follows the principles of common interfaces and consistent definitions that forward-thinking organizations usually adopt during the architecture framework phase of the SBA.

Strategic value

This assessment criterion measures the application's importance in achieving strategic objectives. This should be assessed by users in the context of the strategic drivers as defined in the business context phase of the project.

Upon receipt of the above assessments and after the target architecture is developed, a fourth criterion for assessment should also be applied to the existing application portfolio–that of technical evolution.

Technical evolution

This assessment criterion measures the application's positioning to evolve effectively into the target architecture and to take advantage of envisioned advances in information technology. For example, an application that is written for a hardware environment and language that will become part of the target technology architecture would normally have a higher technology evolution rating than one that is written for an environment that will not be carried forward into the target environment. Likewise, an application that is written in a "portable" language has a higher evolution rating.

Based on the analysis of this data, a summarized assessment can then be developed. These criteria should now be mapped in the following pairs on four-quadrant matrices to allow a high-level determination of the recommended disposition of each application:

- User satisfaction versus Strategic Value
- Technical Quality versus Strategic Value
- Technical Quality versus Technical Evolution.

The combination of this information can be used to generate a summary assessment similar to the example presented below.

For this summary matrix, each application has been classified by placing it in the quadrant that most appropriately represents the combined classifications, also taking into account the likely recommended disposition of existing hardware and system software platforms that currently support the applications (i.e., the likely "technical evolutionary" capability of the hardware/systems software platforms themselves).

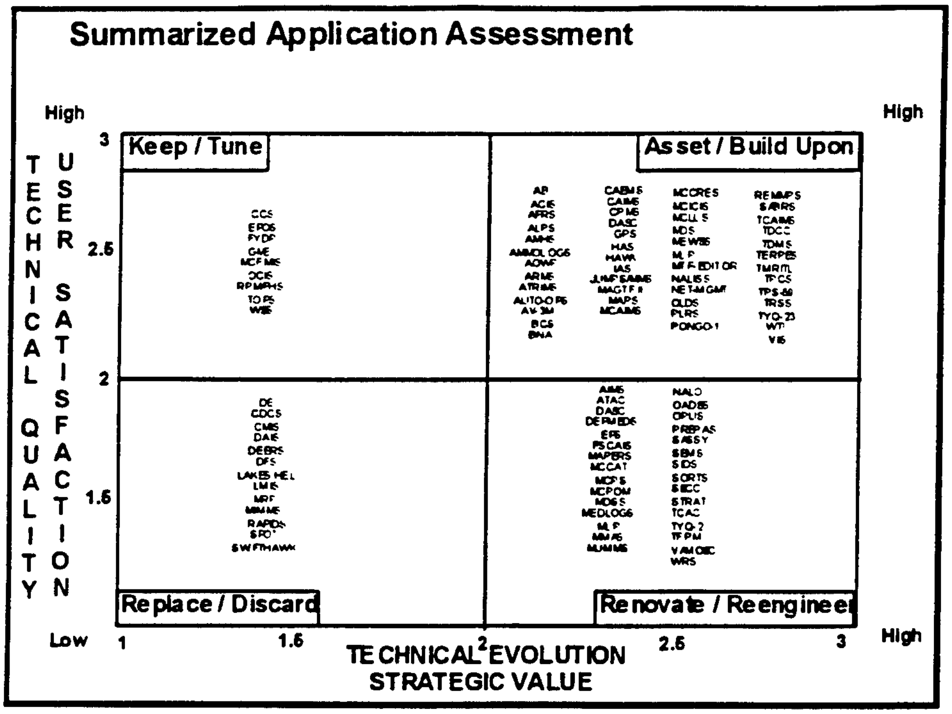


Figure B-3
Sample Summarized Application Assessment

The individual source matrices are typically completed by both users and IS staff within multiple functional areas with input from the client core AWG.

For the summary matrix, each application is classified in a quadrant that most appropriately represents the combined classifications from the source matrices. The following provides a bit more detail on each of these assessment quadrants.

Replace or discard

The application has low user satisfaction, technical quality, technical evolution, and strategic value. If the application is absolutely necessary to the business, it should be completely replaced with a newly developed application or purchased package.

Renovate/reengineer

The application has low user satisfaction and technical quality but high strategic value and operates in a technical environment that can evolve into the target architecture relatively effectively. The application might be given a revamped user interface for improved usability or maybe the programs can be restructured for better reliability and maintainability, perhaps utilizing some reverse engineering tools.

Keep/tune

The application has high user satisfaction and high technical quality but low strategic value and is written in an environment that will be difficult to evolve into the target architecture. Because the application is technically sound and the users seem satisfied currently, keep the application as is for now doing minimal tuning and maintenance to keep it running.

As it reaches the end of its normal life cycle, or as other applications in the new environment have an increasing need to integrate with this application, it may have to be replaced. However, because it is stable and has low strategic value, it should be one of the last applications to be redeveloped or replaced.

Asset/build upon

The application has high user satisfaction, technical quality, and strategic value, and it operates in an environment that can evolve into the target architecture relatively effectively. It should be retained as one of the core applications upon which to build. Applications that fall into the other three categories above should begin to migrate into this category as they are redeveloped, replaced, or converted over the agreed-upon architecture implementation interval.

Rules of thumb

For this summary matrix, each application has been classified by placing it in the quadrant that most appropriately represents the combined classifications, also taking into account the likely recommended disposition of existing hardware and system software platforms that currently support the applications (i.e., the likely "technical evolutionary" capability of the hardware/systems software platforms themselves).

Because these ratings on the source matrices occur independently, there is the potential for a given application to fall in different quadrants on each of the three source matrices (User Satisfaction versus Strategic Value, Technical Quality versus Strategic Value, Technical Quality versus Technical Evolution).

The following rules of thumb should be used in arriving at the summary assessment when independent sources place a given application in more than one quadrant or when the definitions of the quadrants themselves are insufficient to make the determination:

- The strategic value rating on the User Satisfaction versus Strategic Value matrix should be used because this source matrix is completed by the user community (rather than IS support staff). The assumption is that the end user has the best feel for the value of the application to the operational area it supports.

- If an application has low technical quality and low strategic value and low technical evolution combined with high user satisfaction, the application is placed in the "keep/tune" quadrant (i.e., the high user satisfaction and low strategic value combination move the application to the keep/tune rather than replace/discard). The user is happy and it is not a strategic application so, for the moment, keep it going with minimal investment. It will be one of the later applications to be replaced in the new environment.

- If an application has high user satisfaction and high technical quality but low technical evolution and low strategic value, it should be placed in the replace/discard quadrant.

- If an application has low user satisfaction and low technical evolution but high strategic value and high technical quality, it should also be placed in the replace/discard quadrant.

- If an application has high user satisfaction and high strategic value but low technical evolution and low technical quality, it should be placed in the keep/tune quadrant. Over the long term it will need to be replaced because of the low evolution and quality ratings; however, it should not be replaced right away because the users like what they have and it is important to the operation.

- If an application has low user satisfaction but rates high on technical quality, technical evolution, and strategic value, it should be placed in the renovate/reengineer quadrant. The basic application is probably reasonable as a building block, but perhaps it lacks critical functionality or the user interface may be cumbersome. A facelift may be all that is needed to increase user satisfaction.

- If an application rates high on user satisfaction, technical evolution, and strategic value but is rated low on technical quality, it should also be placed in the renovate/reengineer quadrant. The low technical quality is probably of the sort that is not visible to the user, such as unexpected crashes or incorrect data returned. If it were, the user satisfaction would probably not be high. The reason for low technical quality ratings in this case are probably due to difficulty in maintaining, debugging, and enhancing these systems due to poorly structured programs. Given the high user satisfaction, this is an application that should be reengineered internally for more efficient maintainability and execution while maintaining the look and feel it has today.

- If an application is rated low on user satisfaction and technical evolution but high on technical quality and strategic value, it should be placed in the replace/discard quadrant. Regardless of how high the technical quality is in the current environment, if the environment will not be carried forward into the target, the ultimate fate of this application is to be replaced or discarded, depending on the strategic value. In this case, where the application is strategic, the choice will be to replace it with an application that functions in the target technical environment.

- If an application has low user satisfaction, technical quality, and technical evolution but has high strategic value to the enterprise, it should be placed in the replace/discard quadrant. As in the case above, the lack of evolution capability alone is enough to place it in this category. This, coupled with low technical quality in the current environment, provides two compelling reasons to build a new application to support this strategically important functionality.

- Finally, some judgment calls will need to be made where applications end up near the borderline of two or more quadrants. In these cases, low technical evolution ratings should generally pull applications having high strategic value into the replace/discard quadrant.

Appendix C: Detailing the Target Architecture

Most efforts at detailing a target architecture tend to settle on a three-tier model of computing. Each of these tiers is detailed below.

Enterprise tier

It is envisioned that some systems will support virtually all functional areas. In fact, these systems will have enterprise-wide impact through the data they capture and make accessible to users. They will reside at a minimum number of locations (usually only one, but certainly only one within each major area of operation).

These enterprise-wide systems support operations that are common to all work groups. Also, the kind of activities supported do not typically require split-second response times and real-time currency of information, although this may be desirable. The key aspect of systems falling into this classification is that they typically process large volumes of information, and this information needs to be accessed in a consistent way by many users who are usually geographically and organizationally dispersed.

The technology architecture will provide for computer processing to support the enterprise-level systems in a central location(s) with appropriate disaster recovery and security capabilities. All users will be able to access these facilities via network connections.

These systems will probably be positioned to run on high-capacity processors (depending on data volume, response time requirements, etc.). The final decision will be made when the detailed design of the specific application system is undertaken.

Work group

A work group is composed of individuals who share common requirements and needs for information access to perform their function. There are typically multiple work groups within the organization. They typically have a need for quick access to current, function-specific information.

The technology architecture will provide for computer processing in close proximity to the work group to support these quick access requirements. Work group level systems will be deployed to physical locations that support a critical mass of individuals within a work group. Therefore, work group systems may be replicated over the network computing environment.

Within the work group classification will be multiple specialized, but interconnected servers, specifically application servers, communications servers, data servers, etc. These processors will support each individual workstation's need for access to work group data and connectivity to other servers beyond the immediate work group, as well as the enterprise processor(s) that house the enterprise applications.

Individual

The individual level of architecture is the individual worker equipped with a computing platform that is networked to allow access to work group and enterprise facilities. Application systems deployed at the individual level fall into two categories: supporting "tools," such as E-mail, word processing, spreadsheets, and calendaring/scheduling tools; and the "client" portion of work group or enterprise systems, which allow access to data and services that reside on enterprise and work group computing levels. These types of systems may be made available on an individual's workstation to allow maximum customization and autonomy while allowing continued connectivity to other work group and enterprise systems via the network.

Within the work group and individual levels there are further classifications:

Transportable

This is the case of the work group level of computing where one or more work groups are physically transported to a temporary base operation but, once there, they remain fixed for a period of time. An example would be a deployed medical treatment facility in temporary quarters.

Mobile

This is a special case of the work group and individual levels of computing where one or more work groups and individuals are "on the move" and require access to individual, work group, and enterprise computing resources while mobile.

Each level of computing has some unique characteristics in terms of the topology and the components needed to make up the total computing environment at that level. The following is a graphic depiction of each level and how it will interoperate.

To support the location profiles discussed above, the technology architecture has taken the form of a three-tiered network computing environment. This environment provides maximum flexibility to support both common and unique local applications while providing the connectivity required for information sharing. This architecture also provides a measure of local control over systems operation for the various work group locations by allowing critical applications to be co-located with the local work group staff. The three tiers of the target network computing environment are:

Local area networks

A local area networks (LAN) provides terminal and/or workstation access to individual, work group, or enterprise computing resources as well as file sharing and peripheral sharing. The LAN also provides communication with members of the local work group via electronic mail, local office automation tools, and simple localized application systems, which run either on the workstations themselves or on LAN-based processors (referred to as "LAN servers"). The LAN will always provide the link to other network components that, in turn, will link to computer processors. There are exceptions only in cases of deployed mobile computing at the individual level where LAN connectivity is not feasible.

Campus area networks

A campus area networks (CAN) interconnects LANs within a physical work location (or "campus"). Each major fixed physical location will have a single CAN as a "backbone." These fixed physical locations would typically be in CONUS or host facilities that have been provided for long-term usage. CANs will support higher speeds than LANs for rapid message and file transfer between loosely coupled applications that run on multiple work group processors (work group servers) at a physical location or that run on LAN servers as described above.

Workstations and/or terminals will never directly connect to the CAN. These devices will gain access to the CAN only via their LAN connection.

Wide area network

The third tier of the network provides connectivity to a wide area network (WAN) that interconnects all physical locations. The WAN may be a combination of privately owned network facilities including, but not limited to, radio, satellite, and cable; leased lines; and public network services such as Electronic Data Interchange (EDI), packet switching, frame relay, etc. from Value-Added Network (VAN) suppliers. The WAN provides the high-speed, long-haul communications links to interconnect the dispersed physical locations. The WAN provides the capability for applications running on LANs and work group processors on CANs to communicate with remote site applications via message and file transfer or, if necessary, in a more tightly coupled, interactive fashion. The WAN connectivity also allows access to applications that run centrally on an enterprise processor.

Connectivity options

At enterprise and work group locations, LANs will not connect directly to a WAN; instead, they will gain access to a WAN through their connection to a CAN. Workstations and terminals likewise will not connect directly to a CAN; instead, they will gain access to the CAN via their connections to a LAN. Work group processors and enterprise processors will connect into a CAN as well. This allows all workstations and terminals to gain access to all processors via a standard set of network connections.

For the cases of deployed mobile locations, connectivity into the network of computer processors will come through the use of wireless data transmission via a range of wireless technology including, but not limited to, microwave and satellite capability. Depending on the situation, a "traditional" cable-based LAN may be deployed that will be interconnected to the larger community, or an individual computing platform may use wireless LAN technology or individual wireless capability to achieve connectivity directly without a LAN. Anytime such wireless capabilities are used, care must be taken to deal with the issue of security and the possible requirement of not revealing the location of the installation to hostile parties.

Why this computing approach?

This network computing approach with distributed applications and data minimizes the impact of network or processor failure on the enterprise (i.e., the failure of one part of the network), or a local computer will not bring all work group systems down. Also, backup and recovery of the work group that has had the failure can be accomplished by switching it, with minimal disruption, to one or more of the other distributed platforms, if the network connectivity remains intact.

The network computing approach also provides the infrastructure and connectivity required to easily support common services such as E-mail and EDI. These common services have been defined as a required part of many application systems of the future and will be a key enabler to effective information capture and sharing.

Finally, the network computing approach will enable the organization to take advantage of emerging "groupware" packages that allow common work activities to be more effectively automated. Common office automation tools, such as word processing, calendaring, and business graphics, are all more effectively implemented and managed in an environment where connectivity is assured. These work group and individual productivity tools fit naturally on LAN-based platforms. A measure of standardization on these tools and platforms will be necessary for the organization to reap the productivity and effectiveness gains it seeks in the coming years. The network environment will facilitate this process.

Questions to consider when detailing the architecture

There are a number of questions the AWG should ask itself when working through standards at each layer of the DoD reference model. While this is not intended to be an all-inclusive list, here are some questions that a work team can begin to ask when developing the target architecture:

- What opportunities exist for application and technology environment portability within our existing baseline architecture?
- Which of our existing standards meet these functionality requirements?

- What is the impact of DoD standard systems on my functional area architecture?
- What gaps do we have in our standards? Which ones are needed but do not exist? Which ones exist but haven't been implemented in our organization? Why?
- What advantages could be derived through making our current architecture more "portable," more "scalable," or capable of a higher degree of interoperability?
- What kind of benefits are these—cost savings or "value-added" (such as rapid response to wartime situations)?
- What are the "diversity costs" for operating multiple environments across the Logical Operating Unit (LOU)?
- What payoff does standards implementation afford our organization? When and where? What is the business case?
- What is the impact of Federally mandated standards on my functional area's architecture?
- Should we build standards within specific vertical applications, or should we integrate them within specific technology platforms across the organization (e.g., implement standards within a customer service application versus a specific platform area, such as user interface, across the LOU)?
- How much of the existing embedded "legacy" system(s) do we keep? What needs to be replaced? What is the IT and business case for either solution?
- Can we implement these standards? Is the plan realistic? When will we achieve results? What time frame considerations merit review?

Key questions

In addition to the general standards questions, there are specific standards issues to be addressed at each level of the standards-based model. The following questions are presented solely as guidelines. These question sets should be extended by the AWG in every area relevant to the enterprise's architecture.

User interface

- What are the user requirements for user interface (UI) across the functional area(s) with which I am working?
- How global a UI do I want to put in place? Do I want one or several?
- What UI standards do I want to adopt—the X/Windows model, a proprietary-based implementation, or both?
- Will the UI(s) run on proprietary and "open system" workstations? Will they run on both POSIX-based and non-POSIX–based workstations?
- What UI standards does my existing environment support? Can I migrate my current UI environment into a common standards-based set of UIs? How global a UI standard do I want?
- What is the "diversity cost" of the set of UIs in place? Is there an opportunity to eliminate and simplify?
- Which de facto or de jure standards in this area can I make use of now? How standards-compliant are my options?
- What role will my UI play in an overall client/server strategy or cooperative processing architecture?
- If I am not conducting true multiuser/multitasking interactive applications, what is the value of implementing a common UI?
- Is there a suite of "seamless" UI "shrink-wrapped" (commercially available) software available for these "standalone" workstation applications? Neither solution offers the advantages of a true proprietary (VAX, OS/2) or open system-like (POSIX) multitasking environment.
- Does this UI support true multitasking/multiuser work group requirements, or does it really only provide "task switching"?
- Do the UI products I am evaluating support the target platforms I am designing? How do they handle application binary interfaces?

- What will the total cost for my UI strategy be, including costs to upgrade workstations, LAN wiring, implementation, and retraining?

Database

- What are the user requirements for database across the functional area(s) with which I am working?
- What is the "diversity cost" of this set of various databases? Is there an opportunity to eliminate and simplify?
- Which of the de facto or de jure standards in this area can I make use of now? How standards-compliant are my options?
- What is my overall database strategy and architecture? What is the outlook for relational database proliferation? How will functional area(s) handle database related activities in the future?
- To what degree is my current database architecture SQL compliant? To what degree should my target architecture be SQL compliant?

Applications

- What is the scope, depth, and number of the application portfolio across the functional area(s) with which I am working?
- What is the *"diversity cost"* of this set of various applications? Is there an opportunity to eliminate and simplify?
- What will I do with systems that are currently under development but may not be "playing by the standards" I am proposing?
- Which of the existing applications described in the baseline effort should be considered candidates for:
    - Porting to a new open systems environment
    - Recoding into a new environment
    - Redesigning into a new environment
    - Starting from scratch
    - Killing and eliminating?

- How will the existing applications support our target GAE and GTE requirements? How will they coexist with new applications if they are not going to be replaced? What is their life cycle?

- Which applications are redundant? Which of the target applications could be modularly "reused"? Does the code permit reusability?

- What standard programming languages does the application support? Do these languages map to my language standards strategy?

- Are new target applications available in "shrink-wrap" form? What platforms do they currently support? How does my existing and target architecture support these products today *and* tomorrow?

- Are there de facto or de jure standards in this area I can use now? How standards compliant will my target option be?

- What target tools will I use in the conversion process if conversions are deemed necessary? What CASE-tool standards should I use? Do they support evolving CASE standards?

- Does my existing vendor offer porting services for the target application suite in the GAEs and GTEs I am designing? How will I handle conversion if they do not?

- How will new target applications support interfaces to databases, user interfaces, languages, operating systems, communications, management services, and other services? Are there application portable interfaces for these applications in place? What are they? Are they compliant? Are they X/Open XPG compliant?

- What will be the total cost for my application strategy, including costs to migrate and retrain? What are the costs and risks associated with migration?

Languages

- What is the scope, depth, and number of languages in the language portfolio across the functional area(s) with which I am working?

- Have I complied with the DoD policy regarding the use of ADA?
- What is the "diversity cost" of this set of various languages? Is there an opportunity to eliminate and simplify?
- Are there de facto or de jure standards in this area I can make use of now? How standards compliant will my target option be?
- Do I have the professional set of employee resources to sustain and support the existing language requirements? Do I have the right resources to support a new target set of languages?
- How portable is the language(s) and what binary conversion capabilities does it possess? What "degree of freedom" do I have with my existing language portfolio suite?
- What applications and other system components will my existing languages support (e.g., applications, databases, operating systems, user interfaces, communications, management, and other services)? Which one should be:
  - Used in the future?
  - Slowly phased out?
  - Used only for system maintenance?
  - Totally eliminated as quickly as possible?
  - Acquired because we do not have them now but will need in either the short or long term?

Operating system

- What is the "diversity cost" of this set of various operating systems?
- What is the smallest number of languages that I can standardize on today? How many of the languages currently in place do I want to retain in the future?
- Of the languages currently in place, how many of the ANSI-compliant languages have proprietary extensions to them which effectively render them "proprietary" in nature?

- What applications must operating systems support in the target architecture?

- What system calls and operating system standard interfaces do my current operating systems support? What about my target operating systems?

- To what degree will my target architecture operating system environment support standards for network computing? Cooperative processing? Client/server applications?

- What standards framework should I adopt for remote procedure call (RPC)?

- How should my target future operating system handle security?

- How should I integrate new operating systems to be inserted into my existing technology base with embedded systems in place?

- Does the target architecture for operating systems have a migration road map associated with it?

- If I do select a new target set of operating systems that is different than those in place today, will the target architecture support a realistic conversion plan?

Communications

- What is the *"diversity cost"* of this set of various communication systems, platforms, and protocols?

- What target applications should my new communications architecture support in the future?

- To what degree should the new architecture support LAN-based standard environments? To what degree should my new architecture support standards associated with network computing (LU 6.2, DCE, RPCs, etc.)?

- What standards model do I want to adopt in my future architecture? Is there (or will there be) enough product in the marketplace to implement my target architecture?

- To what degree can we implement the OSI model within our new target architecture 1) with existing embedded base products and services, and 2) with new products and services emerging in the marketplace?

- Which of the developing standards (such as X.500) represent significant breakthrough standards that may be of use in more than 36 months (i.e., they are not available for several years but are concepts upon which we would like to establish our target architecture)?
- What new standards, not currently in use, are there that could result in a significantly new way of conducting our functional area's mission, such as EDI (X.12), ISDN or FDDI, or SONET (fiber optic transmission for image and other high bandwidth requirements)?
- What set of target protocols and target services do I want to support in the target architecture?
- If the client/server model is to be implemented in the target architecture, what roles will respective applications have (client or server) to one another?
- What role will the various platforms have with regard to the applications that they run?
- What network management standards do I want my new architecture to support?

Management services

- What is the *"diversity cost"* of this set of various management services located and maintained in different non-compatible environments?
- What is the set of management services that I want in my target architecture? Where should they be located in the target architecture—on one platform or many?

Other services

- What is the *"diversity cost"* of this set of various "other services"? Is the functional requirement cost or opportunity loss of not having certain management services such as access control, authorization, authentication, time, directory, cryptographic, file, data, print, EDI, presentation and monitor/sensor, or actuator? Which of these services should we add, and where should they be located in the architecture?

Application placement within the infrastructure and recommended style of computing

The SBA project participants spent a significant amount of time discussing the descriptions and characteristics of applications and related information subjects in order to provide input on the decision regarding which processor types on which tier of the network would be used to support the applications. The physical location of applications and information can be determined using the technology architecture platform profile described earlier and the cross-reference matrices from other views of the architecture.

Application to Generic Application Environment Matrix

The Application to Generic Application Environment Matrix characterizes each application in terms of the GAEs that will be required to support the functionality of the application. Each target application opportunity cross-referenced to one or more GAEs. This matrix was used as input to the recommendations on application placement across the technology environment. An excerpt from the matrix is shown in Figure C-1 below.

Client/server model

The DoD TAFIM document specifies the client/server model as the preferred standard for distributed network computing. Within the client/server model, five "styles of computing" can be used. Each of these styles of computing has strengths and weaknesses that must respectively be exploited and minimized. The reader is referred to TAFIM Volume 2 for a detailed description of each of these styles of computing. The styles are described below in graphic form in Figure C-2.

Major Business Areas	Applications	Batch Processing	Broadcast	Computer Conferencing	Decision Support	Document Processing	Document Storage & Retrieval	Electronic Mail	Electronic Publishing	Enhanced Telephony	Expert Applications	Hyper Media Processing	Inquiry Processing	Real Time Control	Shared Screen Teleconferencing	Text Processing	Transaction Processing	Video Processing	Video Teleconferencing	Voice Processing	Voice Mail
Establish Direction	THS Agreement Management Application	X				X	X	X	X		X	X			X	X				X	
	THS Guidance Management Application	X				X	X	X	X		X	X				X				X	
	Medical Total Quality Management Application							X			X	X	X				X				
	THS Medical Situation Management Application							X			X	X	X				X				
	THS Medical Options Development & Evaluation Application					X	X	X			X	X	X				X				
Acquire Assets	Defense Medical Service & Materiel Management Application					X	X	X				X	X				X				
	THS Procurement Management Application					X		X				X	X				X				
	THS Assets Positioning Management Application							X				X	X				X				
	Health Statistics Tracking Application	X			X	X	X	X				X	X			X					
	THS Disbursements & Receivables Application					X		X				X	X				X				
	THS Facilities Management Application				X							X	X				X				
Provide Capabilities	THS Medical Transportation Assignment Mgmt Application				X						X	X					X				X
	Theater Medical Site Management Application					X	X	X	X			X	X				X				X
							X	X	X			X	X				X				X
																	X			X	X

Figure C-1. Application to Generic Application Environment Matrix

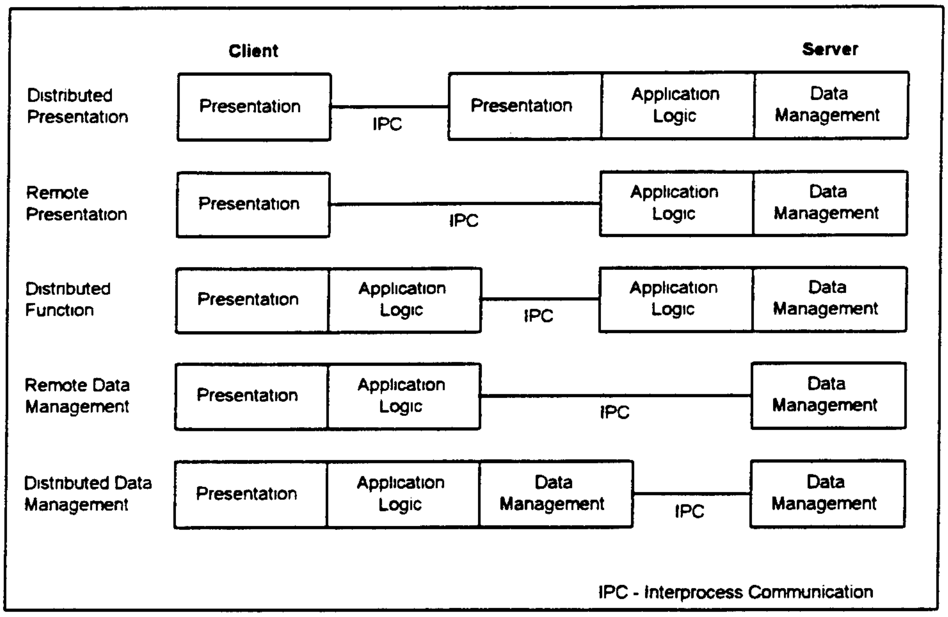


Figure C-2. Client/Server Style of Computing Model

As the above descriptions show, the location(s) of applications and related information is highly dependent on the style of computing chosen for the application and the degree to which a given data grouping (or set of data groupings) is accessed by other applications that may or may not be operating in the same style of computing. Therefore, the logical progression in making these determinations is to analyze each application and its associated information characteristics and linkages depicted in the architecture models and to recommend a preferred style of computing for each application based on these combined characteristics.

The following components of other views of architecture contributed directly to the decisions on the style of computing for each application.

Work view

- Logical operating unit to logical work location
- Logical operating unit to data grouping
- Logical operating unit to application.

Information view

- Characteristics of information

- Information model.

Applications view

- Application description
- Characteristics of applications
- Application to data grouping
- Application to GAE.

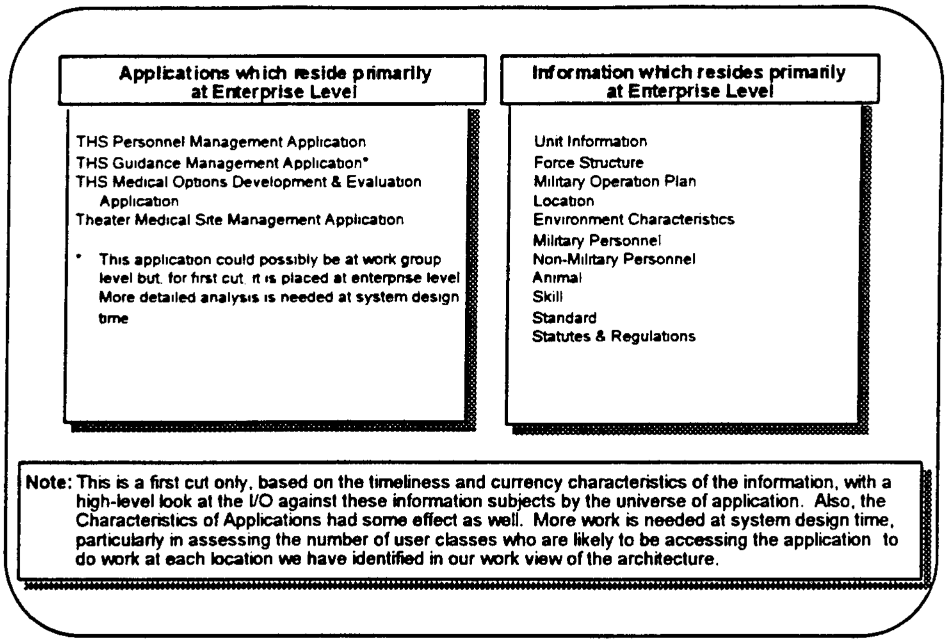
Recommended style of computing and application placement

The following is an excerpt of the recommended style of client/server computing for each target application opportunity.

With these styles of computing in mind, a general mapping of applications and information to the location types was done. Figure C-4 shows an example of the high-level placement of applications and information at one of the three levels within the technology environment:

	Client/Server Style Application	Distributed Presentation	Remote Presentation	Distributed Function	Remote Data Management	Distributed Data Management
Establish Direction	THS Agreement Management System	X				
	THS Guidance Management System	X(1)		X(2)		
	Medical Total Quality Management System	X				
	THS Medical Situation Management System	X(1)	X(2)			
	THS Medical Options Development & Evaluation System	X(1)	X(2)			
Acquire Assets	Defense Medical Service & Materiel Management System		X			
	THS Procurement Management System		X			
	THS Assets Positioning Management System	X				
	Health Statistics Tracking System	X				
	THS Disbursements & Receivables System	X				
	THS Facilities Management System			X		
Provide Capabilities	THS Medical Transportation Assignment Mgmt System		X			
	Theater Medical Site Management System	X				
	THS Personnel Management System	X				
Employ Health	Public Health System			X		
				X(1)	X(2)	X(3)

Figure C-3. Recommended Style of Client/Server Computing Matrix



Applications which reside primarily at Enterprise Level

THS Personnel Management Application
THS Guidance Management Application*
THS Medical Options Development & Evaluation Application
Theater Medical Site Management Application

* This application could possibly be at work group level but, for first cut, it is placed at enterprise level. More detailed analysis is needed at system design time

Information which resides primarily at Enterprise Level

Unit Information
Force Structure
Military Operation Plan
Location
Environment Characteristics
Military Personnel
Non-Military Personnel
Animal
Skill
Standard
Statutes & Regulations

Note: This is a first cut only, based on the timeliness and currency characteristics of the information, with a high-level look at the I/O against these information subjects by the universe of application. Also, the Characteristics of Applications had some effect as well. More work is needed at system design time, particularly in assessing the number of user classes who are likely to be accessing the application to do work at each location we have identified in our work view of the architecture.

Figure C-4. Enterprise Level Applications/Information Model

When a distributed network computing environment is envisioned for an organization, the "location" of the application and information is not definable in concrete terms at the architecture level. By definition, this kind of technology environment will support both distributed data and application processing. Specific instances of a given data grouping and an accessing application within our architecture model may appear at a number of dispersed locations, either through the techniques of replication, fragmentation, or a combination of both.

An example of this might occur in the health care equipment data grouping and the applications that access it, such as the defense medical service and materiel management application. Records containing the data elements that describe a particular piece of equipment may appear on a computer system in the work group where the equipment is in use. However, some specific data elements about this same type of equipment may appear on another computer system in another department, which may happen to have some of this equipment in use there.

Because information can appear in many locations and computer system platforms in a distributed computing environment, our applications and information architecture implementation must support methods of data synchronization and control that are independent of the applications accessing and updating the data groupings.

Other applications and supporting technology platforms

The next area to consider is inter-enterprise connectivity. Connectivity with other systems in the three services, DoD units, and other Federal agencies is increasingly important. The touted benefits of open systems technology (i.e., portability, interoperability, and scalability) can most effectively be used in this arena. With the adoption of open systems (as described in TAFIM), in conjunction with the mission-specific architecture developed in this SBA, the needed building blocks are available to "link" to entities "beyond the boundary," as needed, in an effective way.

The need to evolve to a minimum set with common components

In transportable and/or mobile locations, a key issue is to "economize" the various communications so that they can be routed through an efficient set of voice/data switching and transmission systems. The goal should be to evolve these systems to a minimum set that meets the currently envisioned needs but which, like the networked computing environment with which they must interface, are built using "standard" components or building blocks.

This will not only move the organization ahead in terms of interoperability but also should reduce the number of unique repair parts and end units. This will provide cost and operational efficiency benefits that complement the increased productivity that seamless communications can bring.

Cross-service compatibility is a key issue

The issues of compatibility and interoperability within the body of existing and planned communications are significant. In the joint environment, there are still major issues with mismatches on communications protocols, as well as with system and applications software, which cause severe hardships when joint operations are attempted.

We are reminded that various armed services networks utilize commercial facilities for both switching and transmission to augment private networks. AUTOVON is an example of a service that rides on leased commercial facilities. There are pros and cons to each approach.

When commercial facilities are used, the organization gains something in that the operation of the network is not a burden, and the underlying technologies and services are constantly being upgraded by the common carriers and VAN vendors. However, the use of commercial facilities introduces the need for minimum service levels and a monitoring process to ensure compliance. Also, these commercial networks have many more opportunities for security breaches than do fully controlled private networks. Based on industry experience, however, when traffic does not require specific security considerations and when they are readily available, commercial facilities are an advantage because of operational and feature-related factors cited above.

The major drawback to dependency on commercial network services is that they may not be available in the diverse geographic and political environments within which the DoD may have to operate. If they are available, their reliability may not be guaranteed. These considerations may lead the DoD to rely almost exclusively on facilities and services that are totally under its control.

The need to further investigate capacity and improved mobility

For the DoD to realize the full potential of the networked computing environment defined in this document, the area of mobile communications needs further investment in two areas:

- Additional capacity for gear that is currently effective
- Gear that will provide new capabilities to transmit and receive a significantly increased amount of digital data in wireless mode.

This is an area that needs to be explored in more detail as each application opportunity moves into the design stage.

From the USMC work done earlier in the year, the SBA team has found that for mobile telecommunications two distinct approaches are in use today:

- Deterministic routing (used by the Navy, USMC, and Air Force)
- Flood search routing (used by the Army).

The flood search routing technique is used in the Army's Mobile Switch Routing Telecommunications (MSRT) system. The MSRT system most closely replicates the features and capabilities of the commercial cellular phone network. It allows "full service" while on the move. In this regard, it is superior to the deterministic approach and should be explored as an evolutionary path. Cellular technology is well tested in the commercial arena and is undergoing continual refinement. This should allow the DoD to take advantage of reduced costs and increased reliability and bandwidth in the long run wherever this technology is feasible. It must be recognized, however, that this technology is not as easily established in a deployed environment as the deterministic method.

From the USMC project, it is understood that the Army is the executive agent for all DoD tactical switching. This includes defining, scoping, planning, scheduling, and determining the operational impact of changes to the tactical switching environment across the DoD services and agencies. Other components of the DoD would be well served by assigning resources to work closely with the Army in this area exploring mobile and transportable switching and transmission facilities options for interconnecting its IT computing platforms.

Security considerations

Security should be implemented at a minimum according to DoD directives. TAFIM Volume 2 refers to a number of standards for security implementation. They are:

- Open systems security
- Multi-domain information security
- Multi-channel processing security
- Distributed processing security
- Security management.

Within the specific components of the technology architecture, there will be opportunities to implement various degrees of security. Security can be implemented at the application level, the operating system level, the database management level, and at the external environment (platform/facility) level.

Multilevel security for secured clients and servers in the technology environment, as well as the possibility of network encryption units (NEUs) for secured network nodes, are just a few examples of the areas covered in the referenced standards and guidelines. The following is a high-level view of the components of a secured architecture.

1. Two constructs provide security extensions to the Target Technology Architecture (1) Multi-level Secured (MLS) operating systems; and (2) Network Encryption Units (NEUs)
2. The operating system on each processor provides for multiple levels of security
3. Each processor has two connections to the network, one for an unclassified network address, the other for multiple addresses (one for each level of security).
4. NEUs provide protection of information passing from processor to network and from network to processor.

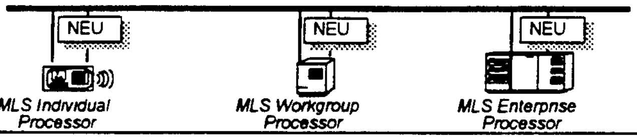


Figure C-5. High-level View of the Components of a Secured Architecture

The reader is referred to TAFIM Volumes 2 and 3 for a treatment of this subject. At a minimum, the DoD should adhere to the standards put forth in these documents paying particular attention to any interfaces between supporting establishment and tactical systems. Of course, the unique nature of delivering health services may actually make the need for security less of an issue than for military operations (i.e., there may be value in identifying a given location as a medical facility).

A look ahead

The next phase in the SBA is the opportunity identification phase. In reality, a significant portion of this phase has been completed during the development of the application architecture view of the SBA.

Migration options follow the opportunity identification phase. This plan will identify and prioritize project initiatives for the next 5 years. The approach will include bundling the projects identified into implementation phases.

Once the project initiatives are grouped into implementation phases, the implementation planning phase begins. These plans will provide more detailed descriptions of the near-term (those started in the first 2 years) projects identified in the migration plan.

When both the migration and implementation plans have been developed and reviewed by the ASC, implementation of the projects can begin. However, the SBA process is not complete until an SBA administration process is defined that will keep the SBA planning process alive and current with changes within the DoD.

This page intentionally left blank.

Appendix D: GAE/GTE/GTP Definitions

Introduction

This appendix contains the definitions of the Generic Application Environments (GAEs) and Generic Technology Environments (GTEs) introduced in Sections 3 and 4. This is an initial set and is not intended to be "all inclusive." Simply put, these should help to get a work team started on its quest to define the necessary GAEs and GTEs for its functional area(s).

GAE sample definitions

Batch processing

Batch processing environments are characterized by their ability to queue work (jobs) and manage the sequencing of processing based on job control commands and lists of input data. The results of this processing include updated information files or databases and often printed reports or special forms that are themselves queued as output jobs.

As such, work is performed asynchronously from the users requesting the job or waiting for its printed output. In most cases, the direct users of the environment are specially trained computer system operators.

These environments have been the mainstay of data processing operations since their inception and will continue to perform critical recordkeeping and background processing functions in conjunction with their related interactive GAEs.

This is evidenced by the major transition that has occurred since the punched card and paper listing days of the sixties. This transition has seen the migration from key punch, through remote job entry (RJE) and optical character recognition readers, to the use of on-line, interactive data entry systems (a transaction processing environment) and inquiry processing systems that share a common database.

Use of file transfers between environments will continue as an effective means of interfacing with batch processing environments, only in a network server context rather than the conventional host computer relationship.

Batch application attributes include number, source, and nature of data capture transactions; timing and sequencing requirements; and volume and type of printing requirements.

Transaction processing

Transaction processing environments support on-line capture and processing of information in an interactive exchange with the user. These typically involve predetermined sequences of data entry, validation, display, and update or inquiry against a file or database.

Environments using character keyboard entry/displays typically base screen designs around the use of menus and electronic forms. Those using GUIs are moving toward the use of icons and images to support command activation and information display.

On-line transaction processing applications have grown out of document processing applications where timeliness and currency of processing a functional area transaction and capturing its associated information is important.

Typical transaction processing application attributes include number, size, source, location, and complexity of transactions; response time; and peak usage requirements. The nature, size, and complexity of associated subject databases (or files) also need to be determined along with the degree of sharing with other applications—as derived from the information model.

Inquiry processing

Inquiry processing environments support functional area activities requiring interactive selection, extraction, and formatting of stored information from files and databases. They are used in conjunction with batch and transaction processing environments to provide information retrieval using either structured (routine) or ad hoc (definable) queries. They are intended to replace the need for extensive reporting systems by providing only needed information on demand.

These environments typically provide user-oriented languages and tools (often referred to as fourth generation languages) to simplify the definition of searching criteria and aid in creating effective presentation of the retrieved information (including use of graphics).

Attributes include frequency of inquiries (prestructured or ad hoc), types and complexity of searching, associated files or databases, and types of presentation required.

Decision support

Decision support environments provide interactive modeling and simulation tools that allow the user to analyze the effects of alternative decisions. These modeling and simulation tools typically work in conjunction with files and databases that were created from batch or transaction processing environments.

As with inquiry processing environments, GUIs are used to simplify the interactions for both building and using the decision support models.

Attributes include the type and complexity of models and simulation algorithms required, the frequency of use, the associated files and databases, the complexity of presentation required, and response time.

Expert systems

Expert systems environments use a type of artificial intelligence built with inference engines and knowledge or rule bases that take or recommend actions based on presented situations and past "experience." They are used to augment human decision-making processes where the "expertise" or thought processes of the decision maker can be defined as rules.

Expert systems are now finding their way into many functional area applications, especially those involving assessment or estimating processes, such as credit risk assessment. These environments are quite specialized today and are based on tight relationships between the "shells," within which relationships are defined, and the corresponding knowledge bases. As experience with applying these environments grows, they will likely become more integratable with other environments.

Attributes involve size and speed of processing, the type of knowledge base used, the type of inferencing processing involved, and whether it is used in batch or interactive mode.

Real-time control

Real-time control environments support event-driven processes supporting monitoring and actuation of physical processes. For this reason, they are often referred to as sensor-based systems. They are designed to handle and process interrupts from a variety of sources (typically involving some kind of sensor device or timer), process associated information through some type of capture or control algorithm, and respond, if necessary, with an appropriate signal to a control or actuation device.

Unlike in the process, manufacturing, and raw materials industries, real-time control environments have a minor presence in financial organizations. They have a role in building security and facility management in such applications as access control systems, fire detection and alarms, energy management, and elevator controls. There are some applications, such as access controls, where integration with other security management environments may be appropriate.

Text processing

Text processing environments support the creation of text documents. They have evolved from the early word processing systems of the seventies to be popularized as part of the explosive application growth of desktop personal computers. They offer greatly improved editing and revision capabilities over the typewriters that they were designed to replace.

Because of their character and word orientation, they offered only limited abilities to improve the presentation and appearance of the final printed document. As a result, they are now losing ground to the graphics-oriented, document processing environments.

Text processing environment attributes include editing and formatting features, mail/merge capabilities, and document filing requirements.

Document processing

Document processing environments extend the basic capabilities of text processing to take advantage of the graphics capabilities of today's workstations and laser printers. Consequently, they provide powerful document and presentation tools for the end user.

These environments use an object-oriented approach to composing documents, allowing the incorporation of graphics, images, and even voice annotation, along with stylized text. They provide advanced formatting and editing features such as style guides, spell checking, use of multiple columns, table of contents generation, headers and footers, and outlining tools.

They require a GUI and often include support for scanning images into bit-mapped representations. This SBA Guide, for example, was prepared using such an environment.

Attributes include types of objects supported, editing, style and formatting features, resolution of display and printing, graphics generation capabilities, color or gray-scale usage, search and retrieval facilities, and document filing requirements.

Electronic publishing

Electronic publishing environments extend document creation and production tools to provide formal publishing capabilities. This includes incorporation of photographic quality images and color graphics, very advanced formatting and style features, such as wrapping text around graphic objects or pictures, and kerning (overlapping characters to optimize spacing).

These environments range from desktop versions to sophisticated corporate publishing systems and are often used through external publishing services. They generally require specially trained "operators" who possess document design and layout skills. They also interface with, or incorporate, sophisticated printing and production equipment.

Attributes include resolution and color; editing, formatting and style features; type, size, and binding of printed output; and printing production rates.

Hypermedia processing

Hypermedia processing is a new environment that extends the object-oriented approach to organizing and displaying information by utilizing various relationships between the stored or created objects. As such, it overcomes the limitation of the printed page and allows the user to "navigate" through the compiled information based on mixed form objects in a manner that is consistent with the needs and capabilities of the user, not some fixed presentation format.

Through the use of the GUI and its extensions to include voice/sound as well as video capabilities, hypermedia presents the ultimate in user communications. In effect, a dynamic document is created by integrating the full range of information display capabilities interacting with associated files and databases under user control.

Attributes include the type and quality of mixed objects supported, the types of relationships allowed, and navigation tools.

Video processing

Video processing environments support the creation of video "productions," either as sequential presentations or as interactive presentations, under user control. They involve both video and sound capture and editing, as well as incorporating still graphics and title generation capabilities.

They are becoming increasingly popular in corporate education as an adjunct or replacement for classroom training. They are also useful for marketing and product promotion or in packaging general information and inquiry services.

Attributes include nature (i.e., analog or digital) and quality of capture and reproduction, editing facilities, ability to integrate user commands, and sequential or direct file access.

Document storage and retrieval

Document storage and retrieval environments are used to retain large volumes of stored information in document formats. Originally these systems were based on microform media using film or fiche with special readers to magnify the information. Computer output microfilm (COM) systems are used to store computer-generated listings or reports.

More recent introduction of optical storage technologies is allowing for storage of scanned or computer produced documents using digital storage techniques. These are now available for use on PC networks as well as for large corporate applications such as archiving. "Juke boxes" are now available to load compact disks under computer control to achieve incredibly high storage and on-line access volumes. Compact disks show considerable promise as a means of distributing reference material with frequent updates possible at low cost.

Attributes include type of media, speed and resolution of scanners, compression techniques, ability to modify or update stored material, access frequency and response, media storage life, and retention volumes.

Electronic mail

Electronic mail environments support the storage and forwarding of directed messages, mail, and other documents or files between sender and one or more recipients. They provide the sender with facilities to create or define the message(s) or file(s) to be sent, use directories and distribution lists for routing information, assign priorities, use preformatted electronic forms, and trace the status of messages sent.

The recipient is typically provided a mailbox with a summarized listing of incoming mail, a log of mail received and read, the ability to file or print mail or documents, and the ability to reply to or forward messages.

These environments are now capable of interfacing amongst themselves to extend their reach from work group to public level (international) distribution. Some are capable of "reading" text messages back via phone access through the use of voice synthesis.

Attributes include sending and receiving features, number of direct users, extent of directory and distribution list management, interconnection capability, and security facilities.

Voice mail

Voice mail environments offer the storage and forwarding of voice messages for a designated set of recipients. They are usually used as an extension of the phone system to provide an alternate to message centers. They typically allow the recipient to retrieve recorded messages remotely from any touch-tone telephone.

Attributes include quality of voice recording, user features, size of directories, and message management facilities.

Enhanced telephony

Enhanced telephony environments provide improved means of using the phone system for interactive audio exchanges between users. Features include call forwarding, call waiting, programmed directories, teleconferencing capability, automatic call distribution (useful for busy customer service areas), and call detail recording.

These can be provided at the local (facility level) or across corporate or public networks.

Attributes include the features supported and the ease of use or help facilities provided through voice response and/or intelligent handsets or integrated voice/data workstations.

Shared screen teleconferencing

Shared screen teleconferencing environments are another newly emerging type of system aimed at supporting more effective remote communications in an interactive mode between two or more users. They combine an audio teleconferencing capability with shared common workstation "windows" that are refreshed on every conferee's workstation whenever someone displays new material or changes an existing display.

In this way, conferees present and discuss displayable material interactively as in a meeting. They can graphically annotate or modify the shared conference window. The attractiveness of this environment is that it can cost-effectively support many of the communication requirements of remote meetings using normal telephone linkages with properly equipped workstations.

Attributes include display quality, refresh and transmission rates, and conference control features.

Video teleconferencing

Video teleconferencing extends the remote meeting environment to include full motion display of events and participants in a bidirectional manner. Thus, the facial expressions and body language of presenters and questioners is displayable to all participants in a conference.

There are a variety of schemes for directing the cameras ranging from fixed position to sender directed to receiver directed to automated sound pickup. This technology has seen limited application because it required studio facilities and was very expensive in its introductory phases. Breakthroughs in charge-coupled cameras, display technology, and high bandwidth communications should see a resurgence in interest and application of video teleconferencing.

Attributes include picture and sound quality, refresh and transmission rates, and camera and conference controls.

Broadcast

Broadcasting environments provide one-way audio or audio/video communications between a sending location and multiple receiving locations. They include the use of private TV facilities that can be purchased or leased for corporate purposes. Many organizations are taking advantage of these facilities and offsetting travel costs for use in corporate announcements and product introductions.

Some information providers are now producing special-purpose TV shows for corporate subscribers as a substitute or adjunct for attending conferences (e.g., *The Computer Channel*). These are often combined with audio return links for question and answer sessions.

Attributes include the quality of production facilities and the scope/range of the receiving network.

Computer conferencing

Computer conferencing environments combine the merits of document creation, E-mail, and conferencing by allowing groups and subgroups to participate in "conferences" via computer workstation. These conferences, however, do not occur in real time. The conferees discuss proposed topics through interacting over time. Conferees, or invited guests, can drop in or out of conferences or subconferences at will. The ability to trace the exchanges is provided.

These environments have become popular among academics and within university circles, beginning with basic text capabilities. Combining the richness of hypermedia with computer conferencing would create an environment in which the most capable and experienced individuals could be brought together remotely to focus on a critical topic using the most powerful electronic means of communicating ideas.

Early forms of these environments are now available to users of graphical workstations. Attributes include types of documents exchanged, conference management and recording facilities, and search and retrieval capabilities.

GTE sample definitions

Each GAE is supported by one or more GTEs. The combination of the GAEs and GTEs provides the infrastructure components for delivering systems and services to the organization.

User interface services

User interface services provide the basic means for users to interact with the computing environment, managing the user interface for any class of user interface device from a simple character terminal to an advanced graphic workstation. User interface services also provide support for the user in navigating through to the appropriate system or server, authenticating the user and managing the user's desktop.

User interface services must support various input and output devices defined in the GAEs for each user class. There will need to be a variety of presentation servers used by user interface services to support the various classes and types of interface devices. For example, there may be an X/Windows-based high-end GUI server and a lower level character-based server for different users.

User interface services interact with all other GTEs providing them with the ability to receive and present information to and from the user. Client applications and users can be reasonably isolated from differences in the underlying technology through the various presentation servers incorporated in user interface services. For example, the user interface should operate in a similar way on a Mac, a PC, or a POSIX workstation.

Optional servers can provide encryption, data, and file management for user interface services. These may or may not be configured into the environment.

System management services

System management services support all activities dealing with the management of the computing environment, interacting with all other GTEs to provide the management capability to monitor and control the total environment.

The objectives of system management services include providing adequate availability and performance across the environment, accurate and complete billing, change control, and failure recovery. This environment provides the basis for implementing specific applications and tools to provide these capabilities.

GAEs and all other GTEs make use of system management services.

Communications management services

Communications management services is another GTE that is used by all of the other GTEs that want to communicate. This environment implements the communications infrastructure consisting of various communication servers, name and directory services for resolving addresses, and authentication and access control for ensuring the appropriate level of security. Thus, all the technology associated with communications and connectivity is bundled into this environment.

Specialized application servers for bandwidth management and other communications functions would also be provided.

Database management services

Database management services consist of the servers required for managing files and data within the technology environment. It consists of data servers that implement databases and file servers that provide local and remote access to various types of files.

Specific application servers may be implemented to isolate the other environments from the physical structure and location of data. Implementation of a distributed data management environment would require a set of specific application servers to support access, manage the physical datasets, and provide the appropriate level of integrity.

Hypermedia

An emerging area for information management is hypermedia. Hypermedia provides a highly flexible way of linking objects. Over time, documents, images, and other objects could be linked in hypermedia databases resulting in the elimination of document management services as a separate entity.

Standards for information management will be required to deal with traditional data management as well as newer technologies for storing other forms of information. Distributed data management capabilities are appearing in vendor's products and need to be addressed through appropriate standards for their usage.

Transaction management services

Transaction management services implement the environment required for managing transaction processing. This environment includes the basic functionality and servers required to implement a transaction processing application. In today's world, CICS would fit under transaction management services. In the future, it is anticipated that a client/server environment will become the norm.

Transaction management services receives requests (transactions) from user interface services and actually performs the transaction processing. It may interact with information management, document management, or distribution management services to update a database or pass the message on to another environment for processing. For example, a transaction generated by a user interface services environment (i.e., a user using a workstation) could link with several environments before the transaction is completed.

The type and nature of the link will depend on the application requirements. For example, the link may be a real-time interactive link requiring completion by the server before the client can do something else or may be a message transfer link where the message or transaction is passed to the other environment for later processing.

This environment consists of authentication and access control servers to control access to transaction processing and at least a data server with which to update or interact.

Document management services

Document management services are analogous to information management services, providing other environments with the means to access and manipulate documents—either text only or some combination of data, text, voice, graphics, and image (a compound document). The key difference between these two technology environments today is the level at which we can manipulate basic elements of information. In information management services, we can access and manipulate each field within the file or database. In document management services, we generally access the entire file or document using application specific formats for manipulating portions of the document. For example, the format for a Microsoft Word document is different from WordPerfect; likewise, the way graphics is stored in each differs.

However, the distinction between the two is one that is based on currently available technologies. Once we have compound document architecture standards and databases that can handle document objects well, it's likely that the two environments will merge and become one.

Conferencing management services

Conferencing Management Services supports the real-time exchange of information from one or more user clients. It permits a user to address a communication to any member of a group without needing to know exactly who is in the group receive communications from all or selected members of the group without needing to know who is currently in the group, and to reply to them in a like manner.

Conferencing services include various types of real-time services including voice conferencing (audio only), video conferencing (audio and video) and computer conferencing (shared screen).

The Conferencing Management Services GTE utilizes Name and Directory services to establish the parties for the conference and is closely linked with the Communication Services GTE to establish the physical linkage. It may also closely link with hypermedia (in information management services) to provide a dynamic subject- and task-oriented asynchronous conferencing environment.

Distribution management services

Distribution management services support the distribution of messages, transactions, files, and any other information between technology environments and physical locations. This environment consists of servers that implement electronic mail, voice mail, and EDI. It also is tightly linked to the communications management services GTE to provide the actual communications between components.

Development services

Development services provide support for all aspects of systems delivery including all phases of the development life cycle, prototyping, and end user development. This environment interacts with the other GTEs to access information on the current infrastructure and to implement changes and enhancements.

Development services is built upon several servers to provide authentication, location of objects (name and directory servers), and to implement specialized applications. CASE tools and compilers are considered to be application servers in this environment.

Repository services

Repository services is an emerging GTE that will provide the repository environment for managing the technology environment and the applications and data stored in the environment. The repository can store information about any "object" in the technology environment including, but not limited to, the physical processors, application modules, data, and processing functions. All of the GAEs, GTEs, components, and servers defined in this document would be entities in a repository.

Repositories for system construction

A passive repository, such as those being introduced by IBM, DEC, and others, can provide the dictionaries and system encyclopedias needed for defining and constructing application systems and data. This type of repository is the essential underpinning of a CASE environment, as it provides the basis for storing information at each phase in the development cycle and transferring that information from one phase to another.

Repositories for systems management

Another type of repository, called the active repository, can be used to store system information and to dynamically manage the IT environment. For example, with the capabilities of an active repository, system management services could manage the execution of applications to optimize performance and reliability.

Conceptually, repository services will interact with other GTEs to provide a "single system image." This is an environment where the computing and network infrastructure appears to the application and user as one "computer." In this environment, repository services would define the single-system image and manage where and how processes are actually executed.

Server definitions

Figure D-1 lists several server types. It illustrates a sample set of logical components of an organization's technology infrastructure. Entries may be added or modified.

Name	Translates network-wide logical names to network address
Directory	Identifies logical names based on attributes
Authentication	Establishes the needed identity of a network user
Access control	Establishes access to desired applications or data
Cryptographic	Provides encryption and key management services
Communications	Establishes linkages for a client (switching, router, gateway)
Time	Ensures common network time
File	Provides transparent access to network files
Data	Provides remote data services (database access)
Print	Remote printing and print management
Mail	Provides electronic mail services
EDI	Provides electronic data interchange
Applications	Provides application-specific services
Presentation	Manages the user interface for a client user (a person)
Sensor monitor/actuator	Manages interfaces to physical sensors, actuators, and timers

Figure D-1. Server Classes

Name server

The name server provides a means of finding an attribute of an entity given the unique name for any entity within the technology environment. Entities can be physical components (computers, workstations, network nodes), logical components (application modules, data storage locations), or users.

The name server will be accessed frequently by clients to find addresses for servers and other objects. Consequently, it needs to be implemented so it can provide high-performance response to queries. The search will be by unique name (unlike the directory server) so quick response can be provided.

Name servers will also likely be highly distributed, so clients cannot assume the attribute provided by a name server is the latest version. While in 99 percent of the cases it will be correct, clients will have to implement a recovery mechanism to deal with the exceptions.

There are few vendor implementations of name servers in the market today. Likewise, the standards bodies are still drafting industry standards for name servers and application programming interfaces to name servers. DEC has an early implementation and architecture for a distributed name service and is worth investigating.

Directory server

The directory server provides a means of finding a set of entity attributes based on qualifiers, such as a telephone number or other descriptive characteristic. Unlike a name server, the searches are often ambiguous and based on a combination of attributes.

Clients may use a directory server in the future for queries such as, "find me a vector processor with 40 MIPs performance" or "find me a storage device with 40 MB free space."

Directory servers will not be accessed as frequently as name servers. Performance will not be as critical as the name server's because of the lower rate of access and the fact that the access by directory server clients is done on an ad hoc-query basis, often under the direction of a user (e.g., find John's telephone number).

Like the name server, clients cannot assume that the attribute provided by the directory server is the latest version. While in 99 percent of the cases it will be, clients will have to implement a recovery mechanism to deal with the exceptions.

The directory server may become a client to a name server to resolve physical and logical addresses.

Authentication server

Validation of users, nodes, programs, and other required objects is performed through the authentication server. Secure channels using encryption and/or some form of trusted communications provide the linkage between client and server.

Access control server The access control server maintains the access control lists for each object within the technical environment. The access control server determines whether access to the requested system object is authorized.

Cryptographic server Encryption services for any process are provided by the cryptographic server. The cryptographic server also manages keys and handles distribution of valid keys among the cryptographic servers. A centralized key management server may be required.

Communications server The communications server forms the basis of managing connections between objects in the environment. It provides connections between objects independent of the physical implementation of the network and ensures accurate delivery of messages between objects.

The communications server, from the point of view of the GTEs using it, provides OSI Level 7 services to the environment. Gateways, routers, bridges, and protocol converters are included in the communications server but are invisible to the clients of the communications server. Bandwidth and capacity management support are also incorporated in the communications server to provide the basis for optimizing the capacity and reliability of the network.

Utilization of this server will provide applications with transparent access to communications services in the environment. The communication server has the ability to support a transparent computing environment where applications and users do not have to be concerned with the logical and physical implementation of the technology.

Time server A critical need in distributed environments is to make sure that time is synchronized throughout the environment. This is especially important in distributed transaction processing applications and database environments where logs need to be kept synchronized to support transaction backout and recovery.

The time server provides time synchronization services to all objects within the environment. Individual objects will call on the time server to get an accurate, consistent time for their use.

There has been limited vendor and standards activity in this area. DEC has proposed their time server to OSF as part of the distributed computing environment request for technology.

File server

The file server provides transparent access to files from workstations and other clients. Unlike a data server, the file server provides access and linkages to the file directories and is not aware of the contents of the file. Processing of contents of a file needs to be performed by the client. The file server does no client-visible manipulation of the data within a file. Essentially, the file server provides the client with the use of a virtual disk drive and little else. For example, in a workstation environment, the workstation would perform all the processing on the file.

This can create synchronization and reliability problems when the file server is used as the place for storing databases and other files that are accessed by several users. The file server is best used when accessing an entire file such as a word processing document or a spreadsheet.

Over time, the file server may be replaced by a data server because of its improved controls and better management capabilities.

Data server

The data server provides data services to clients. A client will send a request to a data server (sometimes called a data*base* server) and the server will respond with the results of the request. The accessing and updating of the data maintained on the data server is performed by the data server, not by the clients.

The data server can provide additional services. For example, recovery and rollback capabilities can be provided. It supports the implementation of better controls by managing access to the data resident within the server.

The data server can also be optimized to the type of data it is being asked to manage. For example, a data server could support archiving and be based on optical storage technology rather than magnetic. In the future, data servers will likely provide access to multiform data that includes voice, text, and image objects as well as data.

Print server

The print server provides common printing services to clients within the environment. The print server provides transparency between the client and the physical printer. For example, differences between different vendors' laser printers should be transparent to the client.

In addition to device-independent printing, the print server also provides queuing, priority management, and other print management services so that the physical printers can be effectively managed.

Printers can be any local or remote output device capable of printed output, including traditional character and line printers, laser printers, fax machines (the printing portion), and even microfilm printers.

Mail server

The mail server provides mail transfer capabilities for a community of users. The basic function is to support the store and forward of interpersonal messages between users. The mail server moves messages based on the contents of the *message envelope* not the message's contents.

The mail server also manages the users' mailboxes. It can automatically acknowledge delivery to a user's mailbox.

The server will support multiform mail transfer (voice-mail, graphics). In the near future, compound mail documents could be transferred using this server.

EDI translation server

The EDI translation server interprets the content of EDI messages and routes them to the appropriate EDI partners. The EDI server works hand in hand with the mail server but needs to interpret the EDI message to translate it or route it to the correct recipient.

The EDI server also provides queue management facilities and assured delivery of messages.

Application server

An application server provides a set of standard application services to clients. It is a form of packaging an application as a commonly used and reusable component of the infrastructure.

The application server must:

- Have a defined application programming interface and message structure
- Be independent of the client
- Provide a set of generic services that can be utilized by a variety of clients (versus a set of services directly linked to a specific application system)
- Hide its underlying process and data from the application—be essentially a black box.

Breaking specific application systems into a client/server model of design is desirable, but the result is not necessarily an application server. The key is to have independence from the client so the server can be utilized by a variety of clients throughout the organization.

Presentation server

The presentation server provides presentation services for a client application and/or a person. It creates a generic presentation environment that is independent of the underlying technology and provides a means for users to interact with the technology environment.

The presentation server is the most user-visible portion of the technology environment. It is the place where the "look and feel" of the organization's infrastructure will be implemented.

Various models and standards for the user interface are available. It should be noted that standards and available products for the user interface are at a very early point in their evolution.

The presentation server will need to accommodate character-based terminals for the foreseeable future, but we expect a migration to graphic-based terminals to occur over time.

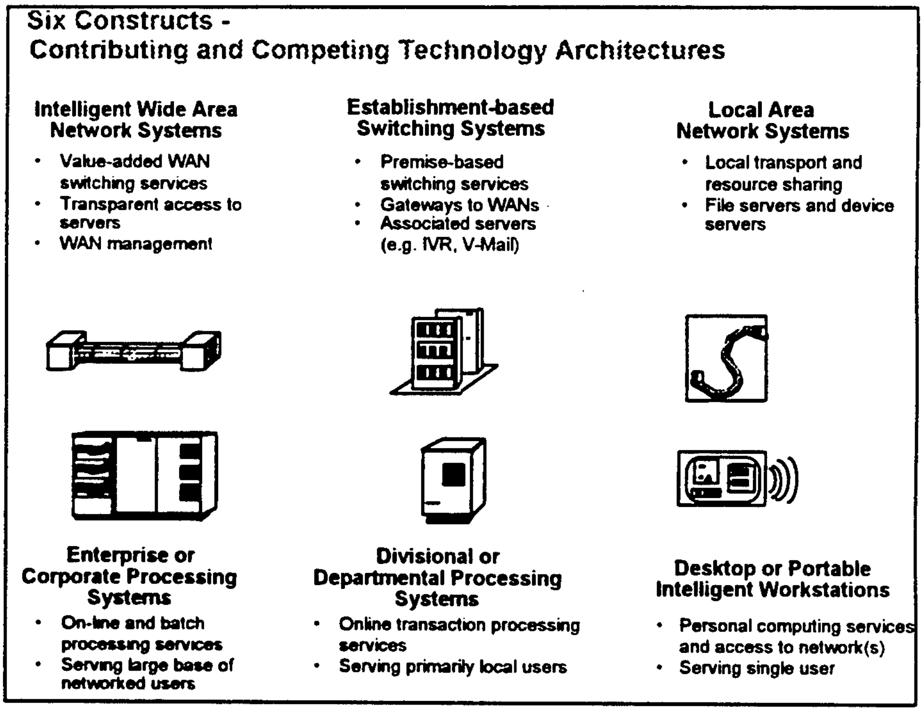
Sensor monitor/ actuator server

The sensor monitor/actuator server provides client applications and users with an interface to physical devices such as cash dispensers, building monitoring systems, or any other device that interacts with physical control systems.

This server is used extensively in manufacturing applications. It can provide the interface to manufacturing equipment, robots, and the like.

Generic technology platforms

There are six technology constructs, or GTPs, are used to provide the fundamental building blocks in a standards-based architecture. Each GTP can function as a fully independent "architecture" in that they each have an interface along with processing, storage, and communications capabilities. As such, each GTP may offer alternative choices in delivery of the same GAE. For example, all six constructs are capable of supporting some form of electronic mail, with different associated strengths and weaknesses.



Six Constructs -
Contributing and Competing Technology Architectures

Intelligent Wide Area Network Systems
- Value-added WAN switching services
- Transparent access to servers
- WAN management

Establishment-based Switching Systems
- Premise-based switching services
- Gateways to WANs
- Associated servers (e.g. IVR, V-Mail)

Local Area Network Systems
- Local transport and resource sharing
- File servers and device servers

Enterprise or Corporate Processing Systems
- On-line and batch processing services
- Serving large base of networked users

Divisional or Departmental Processing Systems
- Online transaction processing services
- Serving primarily local users

Desktop or Portable Intelligent Workstations
- Personal computing services and access to network(s)
- Serving single user

Figure D-2. Six Generic Technology Platforms

It is also important to note that the GTPs do not connote a particular size/capacity. The names for the GTPs connote the usage of the processor, not size. In fact, departmental processors may be larger or smaller than enterprise processors. Some processors acting as LAN servers could well be larger than departmental or enterprise processors depending on the way a given company wishes to organize its work.

Used in combination, these GTPs can be used to describe any architecture environment that current information technology can deliver. Most large organizations are already using multiple combinations of these GTPs. Having determined the appropriate combination of GTPs to support the organization's application requirements, the key to integration is in defining standards that will ensure the highest level of compatibility and portability across the GTPs at both the application and technology platform levels.

Figure D-3 shows a first level of decomposition of each GTP, illustrating the principle components for which standards need to be defined.

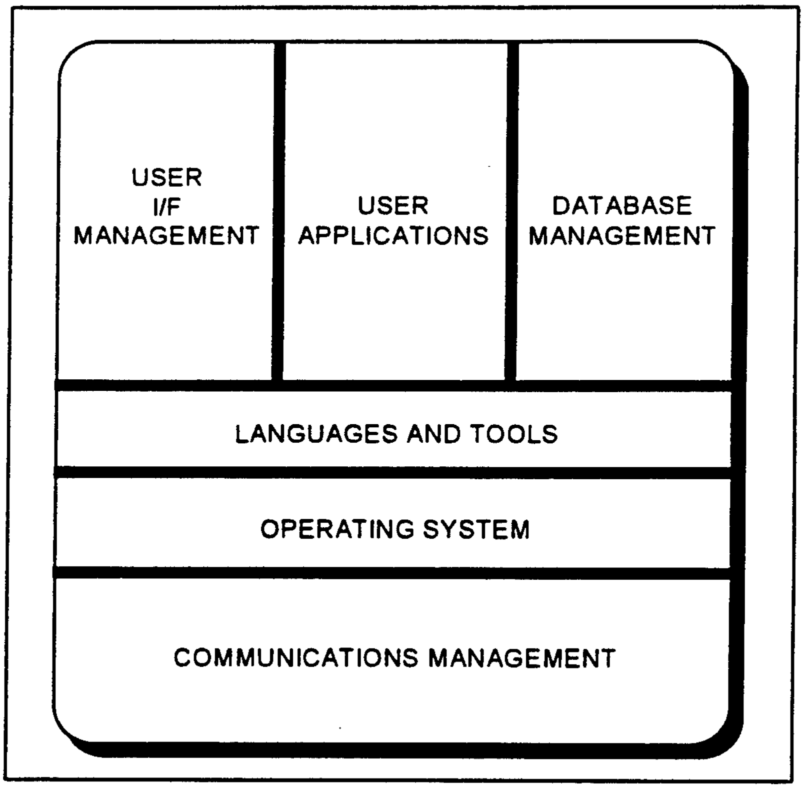


Figure D-3. Components of Generic Technology Platforms

At the component level, we see that all six of the GTPs share a similar structure. Thus, the key to effective integration and sharing in the technology environment is to adopt standards for each component of GTPs, which minimizes the number of different interfaces among components. In today's technology marketplace, vendors are increasingly agreeing on standards at the interface, from GTP to GTP, and within the components of the GTPs themselves. Organizations should adopt technology standards which take advantage of this trend.

This page intentionally left blank.

Appendix E: Migration and Coexistence

The framework for migration

The selection of migration in support of change is always a difficult task and fraught with difficulties and risks. Because the task of migrating information systems and technology is risky, the constraints of migration have to be taken into account in selecting direction and strategy. Many worthwhile projects have floundered because migration was not adequately scoped prior to adoption. In the future, the adoption of open systems and standards-based architectures will reduce the complexity of many migrations to the point where migration will become just one of the scheduled phases, without exposure and without impact on the viability of the strategy. In the meantime, great care is needed to embark on the journey with safety.

As the information infrastructure extends throughout an organization, users draw more and more on the services of a variety of systems. An essential part of migration planning is to accommodate change in one area while accommodating continuity of service in other areas. Coexistence requirements are often as difficult to meet as migration requirements.

What are the migration objectives?

Any migration planning exercise needs to have a clearly defined statement of objective and specification of requirements. In the planning process described, the objectives and primary requirements will emerge from Phase 1, architecture framework, with some refinement of these emerging from Phase 3, target architecture.

For some organizations, the selection of objectives and the movement towards openness will proceed in close cooperation with the development of new functional area systems. For organizations that have a significant investment in infrastructure, or have a multivendor environment, the migration objectives may be very much more technology oriented. Some of the typical migration objectives in the latter category are:

- To move away from dependence on a proprietary infrastructure that has an uncertain future
- To introduce increased interoperability between platforms in the current environment
- To introduce increased openness and integration across platforms in the current environment
- To introduce increased standardization in the current environment so that economies are realized
- To standardize a multivendor environment
- To introduce standards of compliance providing a level playing field for equipment acquisition
- To achieve portability and scalability
- To increase the extent of reuse of technology, applications, and people
- To create an environment that better accommodates new non-proprietary technology
- To introduce new technology
- To facilitate interconnection and interpretability with other organizations
- To work towards the network computing vision within the organization or with other organizations.

Development of these objectives so that they provide clear improvement rather than just a rationalization of costs will flow by examination of key technology issues as they affect the functional area within DoD. Typical questions may define requirements for openness and standardization:

- What interconnection with suppliers is required to improve service/support or reduce costs?
- What interconnection with internal customers is required to improve the service, provide superior products, or reduce costs?
- To what degree can information technology improve or create services?

- Are there particular forms of technology that will change the nature of information processing within the DoD?
- What forms of electronic product distribution (within the DoD) would benefit our functional area?
- What industry-based technology initiatives do we need to come to terms with or accommodate?
- What are the interpersonal communication flows on which our organization depend? What will the benefit of interorganizational electronic mail be?
- What functional area/transaction documents flow with other organizations (outside the DoD)? What benefits would accrue by passing these electronically?

Dilemmas

The evolution of standards is proceeding on many fronts but not at the same pace. The dynamics of standards evolution relate to the complexity of the subject area and the extent of vested interest supporting standardization versus the extent of vested interest resisting standardization. The scene is complicated by the variety of standards bodies and the spectrum of standardization covering de facto standards through to de jure activity.

Of the technology components identified as major building blocks, the most significant level of standards activity is proceeding in the areas of database interface, operating system interface, graphical user interface components, and communications network protocols. In addition, languages have traditionally been an area of standards activity.

The drive for change comes with the attendant problems. While they have been dealt with in some detail in the architecture sections, they remain to be addressed by migration strategies. The dilemmas are repeated here and described in slightly more detail because they have a direct impact on migration.

GUI vs. character vs. block mode terminals

The significant attention given to GUIs flows directly from the level of functionality and ease of use that they can provide. To fully utilize this technology, applications must be modified to support the selected GUI interface.

The conversion of existing character mode or block mode programs to support GUIs requires significant change in program structure and presentation programming. The support of the enhanced functionality requires work to establish and support pull-down menus, pointing devices, and context sensitive tools. The introduction of a GUI approach will, in most instances, require distribution of some part of the application functionality or presentation. Support of distributed function requires an infrastructure that provides services such as program distribution, software inventories, remote diagnosis, and file transfer. These increase the size of the migration activity.

Another area of difficulty is that the selection of a GUI comes with its own set of infrastructure assumptions. Any standards-based initiative reflects its heritage. For example, X/Windows emerged from the character-based segment of the industry. Selection of an X/Windows-based implementation creates a demand for network facilities that accommodate character mode terminals. For reasonable response times, X/Windows needs a local host; thus, the infrastructure requirements may even be in conflict with the needs of character terminals that are currently connected to a remote host.

Selection of a GUI creates a need to examine impacts and migration strategies for both applications and networks.

Peer-to-peer vs. master-slave

A common thrust and assumption in many standards-based activities is that information technology will be deployed in a peer-to-peer manner thus accommodating distribution in any of its many forms. Again, this assumption requires quite a different infrastructure than that used to support the conventional character mode or block mode terminals, both of which reflect a master-slave orientation.

Peer-to-peer connections require a communications network that embodies capabilities such as those inherent in LANs and wide area packet networks. By and large, the WANs established to support block mode terminals are packet based and are thus well suited to support peer-to-peer interoperability. Character mode WANs are unsuitable for support of peer-to-peer communications nor are packet networks able to adequately support character mode applications across the network. Therefore, in this case, the movement to standardization is more easily accommodated within a block mode world than it is within a character mode world.

Database

Another area of significant standards activity is that of databases. The adoption of SQL and the relational model establishes the cornerstone of standards in this area. While standards have established significant standardization in terms of the data interface language, other areas of significance for programming, such as interoperability and distribution, have not received the same attention and do not have communality across the marketplace.

This area of standardization also illustrates the conflicts between standardization and innovation. The emergence of object-oriented databases disturbs the status quo and calls into question the breadth of applicability of the incumbent standards.

Again, converting programs to make use of the relational model is no simple matter. While it is possible to develop migration tools that allow programs with old forms of data navigation to access SQL databases, this does not exploit the capabilities of SQL. To gain the full benefit of the SQL model requires that information be remodeled and that applications be redesigned.

Finding an answer

It is impractical to simply toss a coin when selecting a standard. It is essential that any drive towards standardization be initiated in the context of a well-thought-through architecture for the organization. The trends toward distributed processing and GUIs are immutable. The deployment of these styles of computing needs to be approached carefully by operating within the constraints of available technology and being consistent with the structure of technology placement that matches the long-term direction and shape of the organization.

In resolving these dilemmas, the migration plan will have to adopt a strategy that reflects an assessment of:

- What do we wish to protect and what are we prepared to discard?
- To what degree do we wish to standardize?
- Do we want standards to be vendor neutral, or are we satisfied with proprietary standards?

Taking control and responsibility

Answering questions such as these is, for some organizations, an entirely new activity. For many organizations, the issues of longer term technology architecture and direction are simply left in the hands of the selected supplier. At this stage in the development of standards-based systems, a standards-based policy requires the organization to accept responsibility for its own direction. The organization must clearly understand that it is choosing to pursue its own path through the morass of technology choices rather than simply following the lead of a particular vendor.

Making this decision entails some risk and requires that the organization retains staff with the time and ability to guide the organization. Against these costs will be balanced the benefits that flow from openness. Pursuing this path requires determination and commitment from the entire organization.

Determinants of migration size and complexity

As the scenarios show, the extent of migration activity varies significantly according to the:

- Current architecture
- Target architecture
- Organization size
- Value of technology to the functional area
- Organization complexity
- Extent of change
- Impact on culture
- Cost.

For some organizations, the migration activity may be minor and may not need to be supported by extensive structure and analysis. For these organizations, the extent of planning implied in this appendix may be totally inappropriate. It may be that they can simply "just do it."

For others, the issues of migration and maturity of the standards-based products will be such that, after analysis, the migration costs and issues will loom sufficiently large that the organization will determine that its best interests are served by the retention of a proprietary strategy (at least for the interim—until the costs become less prohibitive).

Baseline characterization

The inventory activities of this phase will provide key information for migration planning on the valuation of existing assets and the identification of risk. From a migration point of view, the necessary inputs may include:

- Valuation of existing investments in hardware, software, applications, development staff, operations staff, users, management, and management process.
- Critical evaluation of existing suppliers, their prospects for survival, and continuity of their product lines. Is the vendor a special-purpose vendor and thus likely to survive in its niche, regardless of standards support?
- An estimate of risk, cost, and opportunity cost relating to the current inventory. For vendors or product lines that may not survive, what is the cost to the organization of loss of impetus as a vendor winds down investment and turns attention to alternative product lines? What is the cost arising from reduced market support? What is the opportunity cost from use of obsolete equipment?

Target architecture — examine alternatives

The selection of a target architecture requires some understanding of migration impacts in order to move towards a practical target. Selection of a target will need to take into account the issues that emerge from the baseline phase while addressing the objectives and targets. Some of the questions that may help the issues emerge are:

- Do we have requirements that can only be addressed with a proprietary-based architecture?
- What is the impact of past investment? What base must be protected?
- What are the general levels of costs associated with different architectures? What is the impact on the total level of expenditure across the organization across time?

Alternative architecture targets may emerge by looking at the organization from various views. Looking at the organization in terms of its functional areas will highlight standardization within a related application set and may subsequently identify pilot opportunities that are not closely coupled with other application systems. Viewing the organization in terms of work organization and the need for application access of each grouping of staff and department will provide input to the needs of the organization in terms of GUIs and integration on the desk.

An inventory-oriented view focusing on the proliferation of platforms will focus on the need for rationalization of platforms and uniformity in infrastructure. Such a view needs to include the network platforms.

A management view of the organization will focus on the integration of information and the needs of the organization.

Opportunity identification

For some organizations, the opportunities for migration will be in the form of specific functional area initiatives with supporting applications. The difference will be that implementation will be based on the adoption of a standards-based architecture. From the functional area point of view, these projects may not represent a significant change from the normal approach of justifying and proceeding with information system implementation. Where such opportunities are limited in scope and proliferation, they make ideal pilot candidates.

For some organizations, open systems adoption will require a gradual modification and migration of the infrastructure. In these situations, there is a significant need for the commitment of the organization to sustain migration over a long period.

Migration options

The evaluation of migration requires that the alternative migration strategies be examined to determine the effort, cost, and adequacy of the approach. This requires research and validation of the elements of each possible migration solution. Typical questions that need to be asked are:

- Is it viable?
- What products does it need? On what standards are they built?
- When will the products be available?
- What can we do to position for future decisions?
- What education and learning must be undertaken?

- How do we introduce the consequent cultural change? What is the cultural change for development staff, operational staff, users, and management?
- What are the relative costs of each option?
- What benefits are delivered by the option?

A caution to the reader

The migration scenarios selected are hypothetical and have been developed for the purpose of illustration only. They do not attempt to portray real life situations. Care must be taken in using the scenarios in that, while the DoD must individually assess its own requirements, the scenarios presume requirements. While the DoD will evaluate migration options based on the latest market knowledge, the scenarios presume the market at a point in time.

The comments and conclusions made about the scenarios are general only, they are not complete. They should not be cast in the light of recommendations. It should also be realized that the solutions presented in each scenario are not necessarily the only ways of solving the hypothetical problems. The investment decision process and relative sensitivity to costs are different for every organization. These scenarios do not provide guidance or commentary on the relative costs of alternative migration options.

Scenario 1: proprietary vendor with a commitment to POSIX

This is a general scenario that covers a medium-sized vendor offering POSIX interfaces to a proprietary operating system as part of a general commitment to vendor-neutral standards. It is assumed that the vendor also commits to XPG and OSI.

In this case, the vendor is committing to comply with the open APIs so that applications written to the standards are portable onto or from their platform. Vendors providing this level of standards support aim to accommodate portability of applications across platforms but have a view that the platform, as supplied by the vendor, is complete.

The alternative view, that standards should be used to allow interchangeable components within generic platforms, has not been considered in any scenario. This concept of openness is not supported by hardware vendors but does receive some support from software vendors and third-party peripheral suppliers.

While every vendor offering POSIX-compliant platforms has a proprietary offering below the interface, the class of vendors represented by this scenario differs from the provision of a POSIX-compliant UNIX in that:

- The capabilities of the proprietary offering are maintained intact within the platform; thus, a single platform can operate in either of the two modes.
- The platform benefits in that the proprietary environment is probably more mature than the UNIX environment. This presumption may not always be correct and will change as the UNIX-based offerings develop.
- The platform will be developed by the vendor in response to two client sets (proprietary and open). It is possible that the proprietary mode will always receive functional enhancement first.
- The development of a new function is limited by the resources of the vendor. The vendor will not normally be able to roll in a function developed by the industry for the UNIX vendors or by the two groupings of UNIX-based platforms (OSF and UI).
- The vendor's solution will not be able to benefit from the ideas of component interchange should the marketplace force vendors along this path.

Current architecture

The current architecture is shown in Figure E-1. The primary characteristics of it are:

- A proprietary CPU running proprietary operating systems with proprietary file systems but with POSIX compliance.
- A platform that is able to include an SQL DBMS.
- Language support that includes COBOL, proprietary languages, report writers, and query languages.
- The platform includes a number of mission-critical applications that operate using on-line update to the databases.
- Normal terminal support that uses block mode terminals, and all applications written to support block mode terminals.

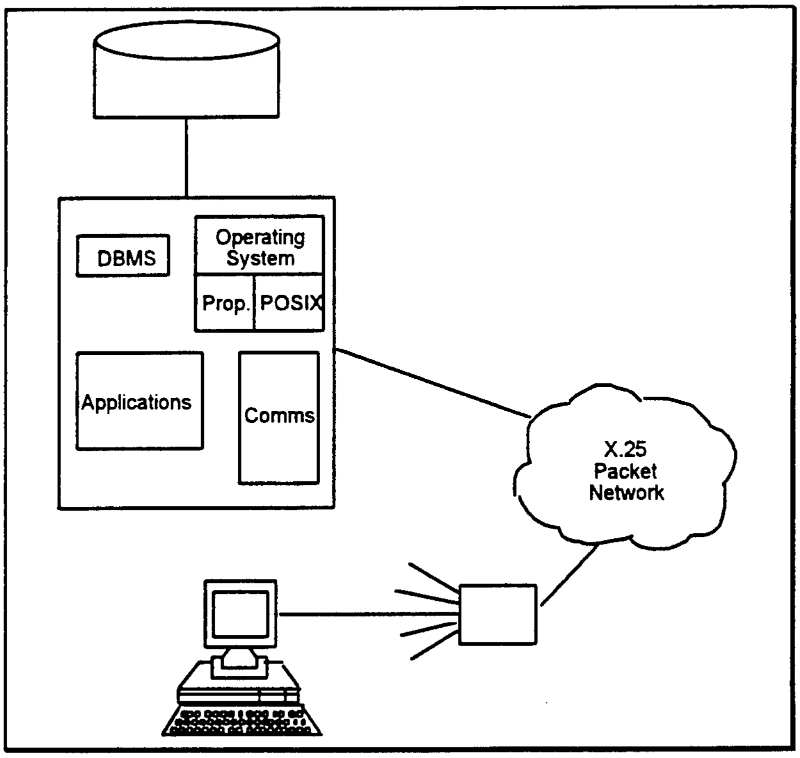


Figure E-1. Current Architecture

- The vendor has committed to support OSI, has an X.400 product in place, and an FTAM product due to be released—it is expected that the vendor will fully support the level 7 OSI protocols a little behind market adoption.
- The vendor has support for X.25, and terminals may access applications via X.25 operating in block mode.

Migration objectives

There is a significant investment in application software; thus, there is a desire to protect this investment. There is no desire to change the user interface for existing applications.

There is a significant investment in block mode terminals. It is required that these be retained for their life rather than be discarded.

There is a desire to use a GUI for some new applications, which creates a requirement for both the GUI and block mode operation to be accommodated.

There is a desire that both old and new applications be able to operate on one platform and share the networks and infrastructure.

There is a requirement that data belonging to old or new platforms be available across both types of applications.

There is a requirement that the software for existing operations, network management, capacity management, storage management, etc. continue in use.

Target architecture

The target architecture requires that:

- The POSIX interfaces be enabled
- The DBMS be SQL based
- The programming language be a standard language
- A GUI be introduced.

Migration options

The scenario assumptions have resolved much of the discussion regarding alternative strategies. The scenario assumes coexistence is available.

Option 1

Leave all old applications intact and write all new applications using the POSIX-defined interfaces. Ignore the need for a GUI and continue to use block mode terminals with the existing networks.

On analysis, this is practical for only a small percentage of applications. Few applications can live within the bounds of the implemented POSIX standards. A number of batch, OLTP, and process control applications cannot operate within the bounds of the available POSIX specifications and/or support. New applications requiring this functionality must use the proprietary facilities.

There are also some conflicts between the POSIX-defined interfaces and block mode operation.

Option 2

Same as Option 1 but also make use of a non-open GUI.

This requires some distribution of the presentation layer. The selected GUI is Microsoft Windows. By using PCs on a LAN with block mode emulation to the host, it is possible to accommodate both the block mode terminal applications and the GUI-based applications, but the GUI is not open.

Option 3

Same as Option 1 but using X/Windows as the GUI from the central host.

This option proved unviable. X terminals were not able to support block mode emulation. Workstations able to support the block mode operation and X terminal emulation could not viably attach through the network to the host-based X applications.

Option 4

Same as Option 1 but using X/Windows and distributed presentation.

Apart from the issues raised in Option 1, the X/Windows-based GUIs are somewhat incompatible with the LAN facilities required to support the block mode terminals. The solution requires that each terminal be replaced by a workstation, with a presentation layer being distributed to the individual workstation. The presentation layer then requests service from the applications in the central host.

By using the existing block mode as the interface, it is possible to use X/Windows over existing applications.

Option 5

Move to OSI network while retaining block mode terminals and supporting X/Windows.

Again, this scenario is only viable where functions can be distributed to the workstation. The use of X/Windows precludes the use of OSI all the way to the terminals. The use of X/Windows also displaces the currently mature network facilities and network management capability.

Preparing for migration

The scenario revealed a number of exposures. The following activities are warranted:

- An assessment of the viability of the supplier. Should the supplier be unable to continue to maintain development of the two product lines, this scenario will revert to be similar to Scenarios 1 and 2.
- An assessment of the vendor's development funding is necessary to determine what confidence there is that new open functionality will be delivered to match the marketplace. It is assumed that the vendor will be prepared to reveal internal information to indicate the viability of the strategy.
- A brief on standards activity is needed to fully understand the complexity of standards compliance in an environment that must also continue to support the proprietary standards.

Preferred migration

The preferred migration option is Option 2 based on the existing network and a non-open GUI.

The improved compatibility with the installed base over the X/Windows options is significant. Given the inability to fully comply with open standards, it is not clear what the benefits are of partial compliance.

The selected approach includes:

- Use of POSIX standards only where the whole application is able to operate within the standard
- The ability to distribute a presentation layer but no obligation to do so for all applications
- The ability to make use of the GUI for new applications but no obligation to do so
- The introduction of standards-based LAN platforms and workstation platforms to replace the existing terminals and cabling system
- The ability to purchase application packages that work to the POSIX interface standards.

This option provides the confidence of staying with the old while being able to watch the emerging marketplace activity in the open arena.

Conclusions

- A migration is not possible without a total commitment to open standards.
- The use of X/Windows does not fit well with the block mode orientation of the vendor.
- The use of OSI requires some distribution of function.
- The movement away from proprietary networks and block mode operation raises some issues of transaction integrity and recovery. Even though a LAN can support these requirements, the devices attached to the LAN may not unless they emulate the block mode operation.

For example, a remote check printing application that requires confirmation from the printing device that printout has completed without a paper jam as a condition of transaction commitment will not be able to obtain the required status advice under several of the migration options.

- The use of a GUI requires some distribution of function.
- The accommodation of distributed and centralized applications is difficult. The mix of proprietary for centralized and open for distributed is difficult.
- The adequacy of the strategy assumes that the vendor will survive.

Scenario 2: complex multivendor installation

This scenario covers a large conglomerate organization having a variety of vendors represented in different parts of the organization. It is assumed that there is a mix of vendors including IBM, Digital, Unisys, UNIX platforms, and PCs.

Current architecture

The current architecture is shown in Figure E-2. The primary characteristics of it are:

- There are no corporate systems. Each vendor's equipment has a reason for being there, but none is seen as the corporate system.
- Each platform has its own network and terminal set. All of these operate in the mode native to that supplier.
- The IBM platforms utilize 3,270 applications with an SNA network.
- The Digital platforms make use of character mode terminals.
- The Unisys 1100 platforms cover a variety of UNIX suppliers. All make use of ASCII character mode terminals and applications. None has an extensive network.
- PCs proliferate throughout the organization and operate standalones as well as in terminal emulation mode for any of the major platforms.

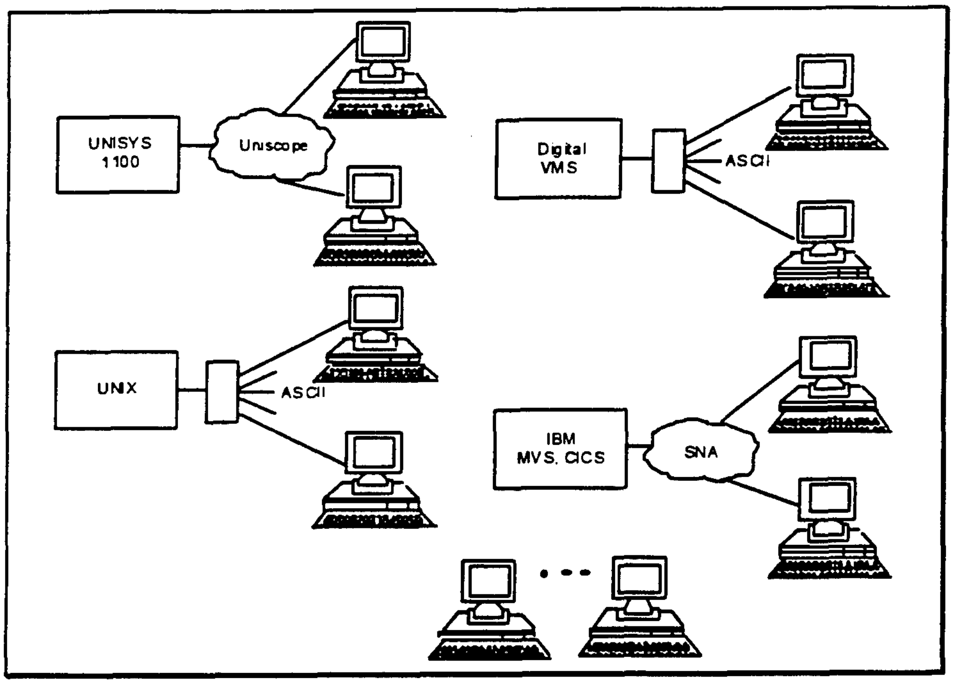


Figure E-2. Current Architecture

- There are no LANs in place.
- There are no shared networks other than at the physical level where TDMs are in place to comb the leased line requirements where these are required.
- Applications cover the range of GAEs including OLTP, interactive computing decision support, office automation, real time, and special purpose. There is no integration of office automation functionality.

Migration objectives

The migration objectives are multiple. None are obligatory, but in order of importance they are:

- Move to a single user interface (preferably a GUI) across the whole organization.
- Move to an environment where any user can access any application.
- Move to a single programming environment so that any development staff can be deployed on all projects.
- Move to a single operational management environment, so that IS operations management can manage the total investment in one coordinated way.

- Move to an integrated information environment where all data can be shared. There is a requirement for both centralized corporate data on the mainframe platforms and distributed work group data on LANs.

The organization has indicated it is willing to redevelop any applications in order to address the migration objectives.

Target architecture

The target architecture is shown in Figure E-3. It is characterized by:

- Multiple mainframe platforms
- A shared network
- A single workstation type able to access applications on all mainframe platforms
- Platforms that provide terminal access from any terminal to any application
- Platforms that provide access to data on any platform from any application or workstation.

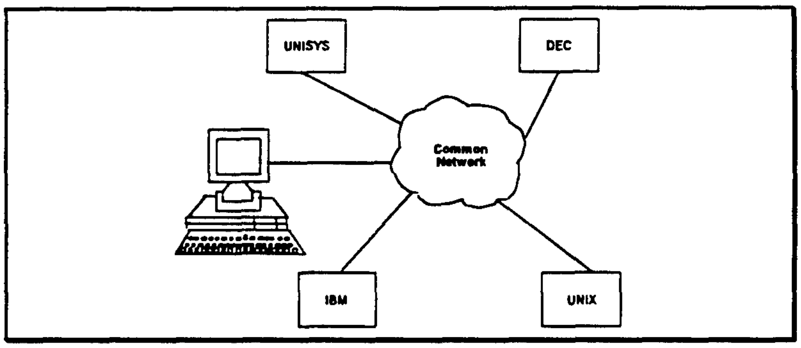


Figure E-3. Target Architecture

Migration options

The alternative migration options are:

Option 1

Implement a shared network that attaches to all hosts and is able to support the variety of terminal types such that any terminal can access any application.

This option proves to be unworkable. The two main problems are the conflict of character mode versus block mode and the need to convert the proprietary protocols and formats.

A network of LANs with bridges and routers can pass character mode traffic in a responsive way but does not accommodate the various protocol converter requirements. Additionally, the cost of the network is significant because of the bandwidth required to sustain responsive network transit. The solution is of doubtful adequacy in addressing the future support of X/Windows unless distribution accompanies the introduction of the GUI. There is no capability of using X/Windows on a broad scale.

While protocol conversion facilities are superior, it is still impractical to provide an "any-to-any" capability. Products are available to support almost all of the combinations, but the technique for addressing the need of each is quite different. In some cases, it requires a back-end solution, while in others it requires a front-end or protocol conversion. Combining them all requires installing some navigational intelligence at the front end and requires significant definitional coordination. Some custom software is needed for ASCII to UTS but can be modeled on available software.

The net conclusion is that this is not a viable approach.

Option 2

Same as Option 1 but convert all character mode applications to operate in line mode with local pad devices.

This approach is assessed as not strategic. It does not move forward; it reduces functionality for some applications and does not facilitate the introduction of a GUI.

Option 3

Review all applications in terms of GAE requirements and work toward a rationalization of platforms by redeveloping applications on fewer platforms

This does not increase openness or integration, it simply reduces the diversity at the cost of significant redevelopment.

Option 4

Redevelop applications on the platform that combine the most mature environment with the potential for future openness. In the redevelopment, use techniques that will ensure future portability, regardless of the standards, through the use of insulation layers and local high-level language facilities. In practice, the selection of a single platform would need to give weight to the extent of the existing investment and the availability of alternative off-the-shelf applications. This option ignores these practical issues for the sake of illustration.

In terms of the GAE functionality covered by the baseline definition, the IBM environment provides the greatest match and maturity. The IBM environment is also supported by all other platforms to some degree or other but mostly acting in 3270 terminal emulation mode. This eases migration phases. The IBM environment is not amenable to open systems development within CICS or IMS.

The Digital environment is assessed as providing significant maturity, particularly in terms of the connectivity options that it supports, while also providing significant opportunity for open development. It combines support for the proprietary solution with OSI and POSIX compliance from within the one platform. It would be the selected platform under this option.

Option 5

Move as many applications as viable onto UNIX platforms and assess the remainder for rationalization onto a single platform. Migrate to a WAN capable of supporting the selected platform's protocols.

Option 6

Move everything to UNIX regardless of suitability and put up with the inadequacies. Implement a standard network and an X/Windows-based window manager. Implement data server functionality across all platforms.

This approach suffers in that it forces distribution onto the workstation in order to get X/Windows functioning. It also requires LANs with TCP/IP for the network with an eventual migration to OSI. These present difficult migration phases for some of the proprietary platforms.

Option 7

Distribute as many applications as possible and, for the remainder, distribute presentation with all the existing platforms being retained as application servers.

This option would permit the implementation of X/Windows with the front-end host then using a variety of techniques for accessing the application servers, including RPC for hosts that support it and terminal emulation for the remainder. This would require that character mode applications be converted.

Option 8

Leave existing platforms, applications, and networks intact but define a new environment with a shared network for use in developing new applications. Over time, the applications will migrate as they reach normal end of life.

The most open new environment is the use of X as a GUI with a presumed distribution of the presentation layer or the whole application. Where access to new applications is needed, a LAN is implemented with access to both the block mode hosts and the new network. Alternative products such as xterm 3270 can be used to provide access from within a window.

The shared WAN does not have to carry either block mode or character mode traffic because these remain on the existing networks. Thus, it can be based on TCP/IP without needing to review enhanced capabilities such as 3270 over the network. It would be possible to run an X.25 network, but this would require TCP/IP to run over X.25 to support the NFS/RPC protocols, which is not preferred.

Existing centralized character mode applications require separate network facilities with access to these from the LAN. Solving the character mode requirements creates a complex solution that is difficult to support.

Preparing for migration

This scenario revealed a number of exposures. The following activities are warranted:

- A critical view of work flow to determine what the real need for integrated access to applications is, compared with the presumed desirability of full integration
- A critical view of management processes to determine what information consolidation is required now and in the future
- A critical view of application-to-application flows, including a forward looking view that postulates future requirements
- A critical view of platform characteristics, GAE requirements, and an assessment of these against the adequacy of products available in the marketplace
- An activity to review all applications to determine the suitability of distributing them to operate within a local work group or to distribute the presentation layer with

application service calls being used to request service from the centralized platforms

- A plan to rationalize the number of platforms over time.

Preferred migration

The preferred migration option represents a combination of elements from the other options and an attempt to get the best of everything. A schematic of the option is shown in Figure E-4. The characteristics of the option are:

- All existing applications remain on the existing hosts. Office automation is to be introduced as a local capability with a corporate electronic mail and document storage/retrieval capability.
- A rationalization project is initiated to reduce the variety of platforms over time. In the meantime, each will be supported with only some modification. It is assessed that resources are better directed to tasks other than redeveloping applications.

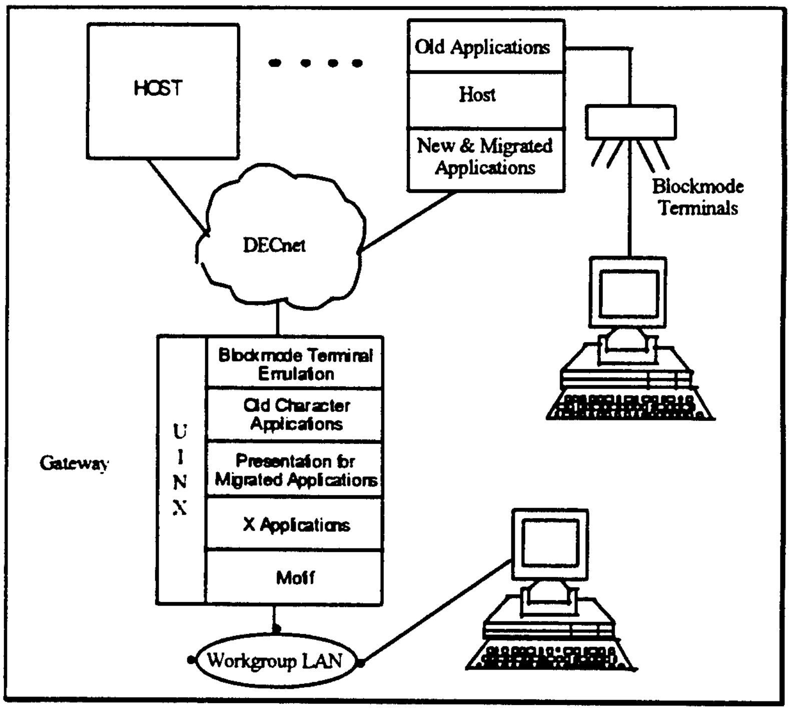

HOST
Old Applications
Host
New & Migrated Applications
Blockmode Terminals
DECnet
Gateway
U
I
N
X
Blockmode Terminal Emulation
Old Character Applications
Presentation for Migrated Applications
X Applications
Motif
Workgroup LAN

Figure E-4. Preferred Migration Option

- All character mode applications will be reworked to become either:

  - Line mode and centralized
  - Character mode and distributed
  - X/Windows and distributed
  - Distributed presentation (X/Windows) with RPC connection to the centralized application server.

- Preferably, new applications will be implemented based on POSIX, a GUI, and standards but, where close coupling exists with existing applications, there may be a need to continue implementation on other platforms.

Thus, the applications will use RPC to access a POSIX host or some form of client server using block mode or other protocols to access application servers operating on IBM, Unisys, or Digital. Access to Digital hosts can be accommodated in either of the above styles.

- Motif has been selected as the GUI of choice given the presence of Digital platforms. It is based on X/Windows. Motif may not be supported by some vendor environments.
- Existing block mode terminals will be retained where possible.
- The standard LAN platform will provide access from workstations to a UNIX-based gateway local host that will provide access and conversion facilities as required.
- A single network is to be established that will carry all traffic. It will be based on DECnet.

Network considerations

The analysis of network options encompasses a review of open networks as well as the use of proprietary networks. It is assumed that, apart from the existing block mode terminals, the network will need to support TCP for the RPC connections to UNIX and DECnet for similar access to the Digital hosts.

The analysis of the use of a neutral TCP/IP network is difficult due to the scarcity of information. TCP/IP networks are often established on a systems-integration-basis with components sourced from a variety of vendors. Carrying SNA and DECnet traffic over IP is understood to be possible, although the product quality is not known. Carrying the character mode traffic is impractical and carrying the UTS traffic requires protocol converting UNIX minis and replacement of terminals with PCs.

The same limitation for character mode also applies to the use of X.25. While some classes of SNA traffic can be carried on X.25 (3270 and PUT4), and DECnet and UTS can be carried on X.25, the strategy is not favored.

If SNA is used to provide the network, a number of shortcomings exist. There is no way of carrying TCP over the network, thus there is no simple means of carrying RPC. It would be possible to implement an RPC transport mechanism based on APPC, but it is suspected that the approach would also need the IBM CSFI product set. Handling DECnet over SNA is also a problem area unless it is transported over X.25 over SNA. The approach is very complicated.

The engineering solution based on shared bandwidth and separation of the logical networks is also not preferred. It involves a significant outlay for additional equipment and suffers from a lack of flexibility.

The selected approach is to use DECnet as the transport mechanism. It provides good support for RPC and potentially supports the TCP/IP protocols as well as accommodating SNA over DECnet in a variety of forms. It cannot accommodate PUT4 but, in this configuration, this is not an issue. Provision of UTS traffic is by using Unisys 3270 support to replace the UTS terminals with 3270s. This has minimal impact on the Unisys applications.

Other considerations

There is a need to control the development of new applications so that over time the organization moves to a more cohesive architecture. The organization is determined that compliance with standards and uniformity across the organization will not be at the expense of functionality and, thus, has a willingness to continue with some proprietary systems where there is a demonstrated need.

A process of architecture review is to be introduced as part of a tighter approval process ensuring that there is a movement towards rationalization.

Conclusions

- The needs of character mode applications and block mode applications fight each other all the way down the line.
- Introducing a GUI requires distribution that will require redevelopment of the application regardless of whether it is character mode or block mode.
- Introducing distribution requires a uniform transport mechanism. Accommodating coexistence creates a complexity of requirements that may be impossible to meet.
- The standards-based approaches represent a particular style of solution. There may be more appropriate solutions, but they may not be open.
- Distribution requires careful planning and analysis. Again, the various open and proprietary products assume different architecture for distribution.
- While a solution on paper has been identified, it is not completely open; and it requires a significant level of validation to demonstrate its viability.
- The questions of operational viability and the adequacy of the selected products in real life still remain to be verified by test laboratories and pilot projects.
- The process requires significant planning skill as well as access to technical planners who are familiar with the products and the environment. Pursuing the selected path will require major commitment from executive management.

Appendix F: Cost/Benefit Analysis

Introduction to a business case analysis approach introduction

This appendix describes the process of performing a cost/benefit analysis (CBA) of information systems alternatives that support a Business Process Redesign (BPR) and systems technology. It is part of an overall economic analysis framework for evaluating the economic effects of one or more subsystems within an object business system.

The object business system can be thought of as an organization, such as the DoD, the Office of the Secretary of Defense (OSD), or a department within OSD and/or a particular work group within the department that transforms inputs into products and services. Furthermore, an object business system can be thought of as a particular work system or set of business processes that are carried out within a particular organizational context, supported by a particular information systems architecture and technology resources.

The DoD has previously implemented an important information management improvement plan known as the Technical Reference Model for Corporate Information Management. This initiative calls for the financial assessment of BPR and information system investments, denoted as Financial Economic Analysis (FEA). DoD guidance on FEA is found in the draft Memorandum for IRM Points of Contact, Budget Bulletin Number 92-04.

The overall approach for performing a CBA applied to BPR and standards-based architecture planning is discussed with the help of an example.

CBA is a systematic financial procedure for evaluating the costs and benefits of an investment opportunity. The investment opportunity may include changing an organization's work system or business process, information systems technology, and/or work group resource assignments. It provides the financial information necessary for management to make decisions about the benefits of adopting new business processes, information technology, and work group arrangements in order to improve productivity, accuracy, timeliness, and reduced life-cycle costs.

Performing a CBA for a new system architecture is a complex task, especially when combined with corresponding required changes in the business process and work groups. It is a task that involves defining the baseline costs for the current object business system, and assessing the potential effects of possibly applying different technologies, different standards, different applications, different human resource assignments, different business processes and different levels of technological experience to successfully satisfy the mission of the organization. In this appendix we have:

- Defined the business baseline
- Defined the technology baseline
- Defined the financial and standards criteria
- Ranked the alternative system architectures
- Presented the key elements in performing a CBA
- Presented the key financial measures and risks.

Determining the business baseline and benefits

CBA focuses on the evaluation of alternative investment strategies and management practices aimed at improving user and management productivity and reducing life-cycle costs. A different analysis compares current baseline operational and management costs with the expected costs for one or more investment alternatives.

The framework for analysis in determining the business baseline and benefits is found in Figure 1-1. The process begins by first defining the object business system and scope of the analysis. The object business system in this section focuses on a particular business function.

Second, a functional analysis of current work activities is performed, and the time and costs for performing the work is collected and analyzed. In addition, output volume, work flow times, technology used, and resources allocated to perform the functions are analyzed. From this information, a Cost Breakdown Structure (CBS) is derived that classifies costs according to a life-cycle orientation. The life-cycle costs may be transformed to fixed and variable cost elements to support the FEA requirements.

This process can be very time consuming, especially when the costs for the activities are not recorded in terms of fixed and variable costs. Therefore, the data may need to be converted by an approximation method with reduced accuracy. Costs are then summarized into their life-cycle phases for activities to provide a cost profile for the work processes.

Third, alternative work processes are identified in order to improve overall productivity and reduce costs. This may include new work flows, activities and tasks, and possibly work group rearrangements to support the updated business processes. The fixed and variable cost structure for the alternatives are estimated with corresponding risk. At this time, the business requirements for standards-based systems, applications, and networks may be identified at a high level.

Fourth, a pro forma estimate of benefits and costs for each alternative is prepared. The costs are estimated for each alternative over the useful life of the systems (e.g., 5 years).

Fifth, the CBA for each alternative is computed with associated risk factors for each alternative. The CBA provides a financial profile of effectiveness measures in terms of their cash flow equivalencies. Costs and benefits are equivalent if they have the same effect. Cash flow equivalence compares the costs and benefits of alternatives in the same terms consisting of: (1) the amounts of the sums, (2) the time of their occurrence, and (3) the interest rate. Interest formulas provide the time value of money viewpoint as a standard for comparing alternative investment proposals. The future amount of a sum can be calculated using the compound interest formula (1):

$$(1) \quad FV = PV\,(1 + i)^n$$

where the Present Value (PV) represents the current or present sum of money, and FV represents the Future Value, given a rate of interest, *i*, for a period of *n* years. The present value (PV) of a sum for *n* years for a given rate of interest can be easily determined by solving equation (1) for PV. This relationship is applied to assess the PV of benefits for the business process alternatives and system alternatives illustrated in the following examples.

Example 1: logistics support services

A BPR study team is assigned to logistics support services to improve productivity. This function is performed across multiple departments. The function is responsible for supplying and maintaining electronic spare parts at selected sites to support the mission of the department.

The cost summary in Figure F-1 represents the baseline costs for the current business process at three sites. The total combined baseline cost breakdown, human resources, and output for the function at three locations is summarized as follows.

COST ITEM	SITE 1	SITE 2	SITE 3	TOTAL
Personnel	$24,000	$50,000	$100,000	$174,000
Transport	$5,000	$10,000	$20,000	$35,000
Facilities	$6,000	$60,000	$60,000	$136,000
IS Services	$5,000	$30,000	$40,000	$75,000
Total Cost	$40,000	$150,000	$230,000	$420,000
Number of Staff	160	1,800	2,000	3,960
Number of Parts Shipped Per Year	220,000	900,000	1,000,000	2,120,000

Figure F-1. Summary Baseline Cost, Personnel, and Output ($ in thousands)

Figure F-2 shows the summary baseline costs per part serviced and maintained. Each person services and maintains, on the average, 535 parts. The service and maintenance cost breakdown per part includes:

• Average personnel costs per part	$ 82.08
• Average transport cost per part	$ 16.51
• Average facility cost per part	$ 75.47
• Average IS services per part	$ 35.38
Total unit cost (rounded)	$209.43

COST ITEM	SITE 1	SITE 2	SITE 3	TOTAL
Personnel	$109.09	$56.56	$10 00	$82 08
Transport	$22.73	$11.11	$20.00	$16 51
Facilities	$27.27	$66.67	$70.00	$75 47
IS Services	$22.73	$33.34	$40,000	$35 48
Total Cost	**$181.82**	**$167.68**	**$140.00**	**$209.43**
Parts Serviced Per Person	1,375	500	500	535

Figure F-2. Summary of Baseline Costs Per Part Serviced and Maintained ($ in thousands)

Figure F-3 summarizes the cost alternatives of two business process alternatives compared to the baseline business process. The PV cost (rounded) for each alternative is:

• Current baseline (PV)	$1,677 million
• Alternative A (PV)	$1,599 million
• Alternative B (PV)	$1,841 million

	COST ITEM	B.P Current	B.P Alternative	B.P Alternative
		Baseline	A	B
1	Annual Recurring Cost		$150,000	$336,000
2	BPR Investment & Migration Cost	-0-	$1,000,000	$500,000
3	5-Year Present Value (PV)	$1,676,934	$598,905	$1,341,547
4	PV (2+3)	$1,676,934	$1,598,905	$1,841,547
5	NPV Benefit	n/a	$78,029	($164,613)

Figure F-3. Business Process Redesign (BPR) Alternative Benefits Compared to the Baseline ($ in thousands)

The PV represents the current or discounted value of a set of recurring cash flows for a predetermined interest rate (8 percent) over a period (e.g., 5 years) plus an initial investment cost for migration. This concept is based on the idea that a sum of money in the future is worth less than that same amount in the present.

The annual recurring cost for the baseline case is $420 million. Discounted at 8 percent, plus the migration costs of $0, gives the PV cost for the baseline case $1,677 million (rounded) over 5 years. Likewise, the annual recurring costs for Alternative B is $150 million. Discounted at 8 percent, plus the migration cost of $1,000 million in the first year, gives a present value of $1,599 million (rounded) over 5 years. Thus, Alternative A costs $78 million less to implement than the baseline over a 5-year period. Similarly, Alternative B has a higher cost than the baseline and therefore is the least attractive financially.

Expected 5-Year Equivalent Costs				
			Risk	Risk
			(-10%) A & B Low	(+10%) A & B High
Work Process		Percent		
Baseline	$1,676.934	100	$1,676,934	$1,676,934
Alternative A	$1,598,905	95	$1,439,015	$1,758,796
Alternative B	$1,841,547	110	$1,657,392	$2,025,702

Figure F-4. Risk Adjusted Cash Flow Equivalent

Defining technology baseline

The process of assessing the benefits of alternative investments in systems and architectures begins with defining the scope and business objectives for technology change. The need for technology change can involve many factors. There may be a need to improve user productivity for accessing data or applications. The need may involve improving development efficiency or promoting portability and interoperability among several systems. Finally, the need may involve improving the operational efficiency and effectiveness of networks, or subsystem components, or reducing the life-cycle costs of systems and/or applications.

Once the scope and objectives are defined, the next step is to determine the target object system and baseline operational costs associated with using and maintaining information technology. This process involves identifying the operational costs for maintaining the hardware, software, and applications. Also, it may include the cost of database access and conversion, the cost of maintaining networks and paying for communication line charges, and the cost for vendor support services including training.

Once the baseline costs for the object system are collected, the next step is to define alternative architectures and systems that meet the business, technical, and organizational requirements and objectives. The system acquisition, operational cost, and utilization cost for each alternative must be collected and analyzed. The initial investments (acquisition costs) for each alternative need to be determined. This involves collecting all non-recurring costs for acquiring, installing, and making the systems ready for productive use. Some of the costs may be fixed charges such as hardware and software maintenance and reuse. Others may vary with the level of system use (variable costs) such as conversion costs, communication access and usage charges, and database storage costs.

Define financial criteria and review open system standards criteria

Prior to performing the cost/benefit analysis and determining cost saving alternatives for the alternative architectures and systems, the financial and standards-based architecture criteria need to be defined. The financial and standards-based criteria need to be incorporated into the business case analysis to support the overall decision-making process. In addition, a method for assessing the degree to which alternative systems support the agreed-to criteria needs to be established. The financial criteria may include cost, productivity, quality, and degree of conformance to standards-based system criteria.

The financial criteria for classifying costs need to be defined. This can have bearing on the overall result. Costs can be classified into their fixed or variable components. Costs can also be classified as direct and indirect, as recurring and non-recurring, and as sunk or past. The fixed and variable costs are based on a level of activity. Those costs that do not vary with the level of activity are called fixed costs; those that do vary with activity are called variable costs. Examples of system costs are fixed disk storage drives, terminals, and workstations. Examples of fixed costs are maintenance costs, depreciation, insurance, and interest on capital equipment. Variable costs are ordinarily defined as those costs that vary in some relationship to the level of operating activity, for example, the network line usage charges, package software and license fees, network support service charges, and computer supplies. Direct costs consist of three components: direct materials, direct labor, and direct expense. Indirect costs consist of indirect materials, indirect labor, and indirect expenses. The prefix direct refers to the fact that the materials or labor used under this classification can be directly associated with the output produced or service delivered, whereas indirect costs cannot. The labor costs for performing the functions or work processes are considered as direct costs. Fringe benefits costs for management services are indirect costs. Both cost classifications are useful; however, when indirect costs are large, the fixed and variable cost structure is preferred. Recurring costs refer to those costs that occur again and again or at specified intervals; for example, the cost of network support services, systems performance analysis, and/or management services activities that all occur throughout the system life cycle. Non-recurring costs refer to "one time" costs that are not repetitive, such as system installation costs, application design and development costs, and application conversion costs.

In addition to cost, productivity and quality standards need to be specified. Productivity is a measure of how well resources are combined and utilized to accomplish specific, desirable objectives or results. It can be thought of as the ratio of results achieved to the resources consumed. The total results achieved are called effectiveness. The total resources consumed are referred to as efficiency. Quality is defined in terms of what is wanted and when it is needed. The "what" is the means for providing the end user with outputs or service that accurately match requirements and expectations. The "when" implies providing users and customers with what is needed on a timely basis; therefore, standards for quality are measured in terms of accuracy and timeliness.

Likewise, the criteria for open system standards need to be established for a given standards-based systems project. The degrees to which interoperability, scalability, and portability are specified in the system requirements need to be determined. Interoperability focuses on the communication methods between machines that provide accurate and reliable transmission of data without affecting the applications that are running. This is the requirement for access or interconnection. It also includes the requirement for distributing or sharing the applications and data across that network. This need to connect, distribute, and share software is the requirement for interoperability. The portability standard addresses the need for application software to be able to run on a variety of computer systems without any work on the part of the user and without any changes to the software. All versions of the software are identical, and the output is readily usable on other machines. Scalability refers to the ability of the same application software package to run with accepted performance on systems of varying size, from microcomputers to minicomputers to mainframes. The degree to which open system standards are represented in alternative systems needs to be established and assessed. The standards for evaluation are found in the Technical Reference Model for Corporate Information Management. The criteria for evaluating standards in this model included level of consensus, product availability, completeness, maturity, stability, de facto usage, and problems and limitations. The standards that are being considered or required to support alternative architectures under consideration need to be ranked for each system alternative. A method for performing this qualitative assessment is shown in Figures F-5 and F-6.

	Standard	Weight		Architecture	
		(1-5)	Baseline	A	B
1	OS/POSIX	5 points	1	8	3
2	Network GOSIP	4 points	8	8	8
3	SQL DB	4 points	8	8	8
4	Languages ADA	3 points	8	8	6
5	User Interface X/Windows	3 points	1	2	6
	* Eight point evaluation scale: 1=lowest, 8=highest.				

Figure F-5. Relative Ranking of Standards-Based Architectures

	Selected		Architecture	
	Open System Standard	Baseline	A	B
1	OS-POSIX	5	40	15
2	Network-GOSIP	32	32	32
3	SQL /DB	32	32	32
4	Language-ADA	24	24	18
5	User Interface X/Windows	3	6	18
	Total Points	96	134	115

Figure F-6. Rank Score of Standards-Based Architectures

Rank and prioritize alternative standards-based technologies

Alternative systems and standards are assessed using a relative ranking method to arrive at a figure of merit. The alternative systems under consideration are matched against the selected standards. A weight is applied for the specified standards. The baseline and alternative systems are assessed on a scale of one to eight (see Figure F-5). The weighted scores are compared to the baseline score (see Figure F-6). This process is illustrated in Example 2.

Example 2: baseline architecture

This system supports the current work process in Figure F-1. It is a large mainframe proprietary computer by one of the leading computer manufacturers. It supports applications. The system supports an SQL database. The WAN and LAN use GOSIP with over 200 active terminals. (Note: Federal agencies are no longer required to use GOSIP; the protocol is specified here as an example only.) The current user interface is propriety and not compliant with X/Windows. The vendor has no plans to meet this standard.

Alternative Architecture A

Alternative A is a multiple minicomputer-based system that supports over 2,000 terminals and personal computers. The operating system is propriety but POSIX compliant. The WAN and LANs support GOSIP. The data based on both systems support SQL, although some vendor options have been implemented. The programming languages are ADA, FORTRAN, and COBOL. The propriety graphic user interface (GUI) is partially compliant with the X/Windows user interface.

Alternative Architecture B

Alternative B represents multiple client/server systems that each support 640 personal computers and over 1,360 workstations. The system has a propriety UNIX Operating system. The WAN and LANs support GOSIP. The database system supports SQL. Languages supported are COBOL, BASIC, C++, and FORTRAN. There is a GUI, but it is not fully compliant with X/Windows.

In summary, the relative ranking in Example 2 of the alternative architectures indicates that the multiple minicomputer architecture (Alternative A) ranks the highest in terms of standards compliance with an index number of 140. Second is the client /server architecture (Alternative B) with an index number of 120. Alternative A is 20 points higher than alternative B as compared to the baseline case of 96 points (index 100).

Perform economic assessment

To perform the economic assessment, we need to include all the costs in each phase of the system life cycle. An overreaching goal of the life-cycle cost (LCC) process is to develop high-quality standards-based architectures and systems based on response to established need. In the DoD, this means deploying standards-based architectures and systems that are competitive in performance, quality, and LCC. The generic LCC model should be applied to assessing the costs of systems from the acquisition phase through the utilization phase. The system's life cycle begins with the identification of need and extends through system planning, systems analysis, systems design and construction, installation, evaluation, acceptance and functional use, maintenance and support, system reuse and, ultimately, phase out. The process represents the life-cycle activities of many systems projects. Although these activities may vary somewhat from one open systems architecture program to another, it reflects a common process for all.

The LCC for each alternative needs to be organized into a Cost Breakdown Structure (CBS). The CBS is a top-down structure that links objectives and activities for each phase of the systems project. It forms a logical subdivision of costs by functional activity areas and major phases. All life-cycle cost elements are considered and identified in the CBS. The costs are coded and entered into a cost/benefit model or database and serve as input to the cost/benefit analysis.

Once the costs for the system alternatives are determined by CBS, the costs and benefits for the alternatives are analyzed and the financial measurements can then be computed. This involves determining the acquisition and utilization costs over the useful life of the system, coinciding with the planning horizon of the organization. Once the costs have been determined over the useful life of the system, the costs and benefits for each alternative can be calculated. In addition, a level of uncertainty can be assigned to the cost elements in the cost/benefit analysis model. A risk assessment can be performed to provide management with a range of benefits that are most likely and least likely to occur. The result of this cost/benefit analysis is then documented and reported to management for decision making.

Summary of financial measures

Assessing the costs and benefits of alternative systems can be represented by one or more measurements using the cost/benefit model. The most commonly used measures are payback, internal rate of return (IRR), and net present value (NPV). A sensitivity analysis can also be performed to determine the range of risk and benefits given a set of risk factors. The payback measure indicates the average of the number of months or years a systems project can take to recover its initial investment. The initial investment usually represents the total cost of acquisition or the cost for planning, designing and implementing, and making the system ready for use.

The IRR is the rate of interest the systems project earns over its useful life. It is the interest rate that makes the equivalent discounted costs and benefits equal; the higher the IRR, the greater the benefits delivered by the systems project.

The NPV calculation represents the net cost equivalent or discounted cash flow value for a systems project. It is one of the most reliable outcome measures of the cost/benefit analysis and is illustrated in the examples that follow. The initial investment costs are subtracted from the sum of the discount cash flows to provide the NPV or net benefit. The NPV takes into consideration the time value of money over the useful life for each systems alternative under consideration. It transforms the costs and benefits for each year into a present equivalent form for comparison. Selected risk factors can then be applied to each of the costs in the CBS. The NPV is then recalculated to produce the risk-adjusted NPV.

The use of sensitivity analysis provides an expected range of benefits, such as optimistic, most likely to occur, and pessimistic. The analysis is performed by assigning probabilities to the CBS for each system alternative. The risk-adjusted NPV provides a level of confidence for decision making.

The initial investment costs, or acquisition costs, are the costs for getting the systems project started, such as acquiring hardware and software. Additional examples include the contract price, shipping, installation costs, license fees, and conversion and/or migration costs. The initial investment costs are the one-time, non-recurring costs for acquiring and implementing system solutions.

The criteria for performing the financial analysis include:

- Agreeing on the cost classification to be used to collect the cost data
- Determining the economic life of the alternative systems and architectures
- Determining the discount rate or time value of money
- Agreeing on the financial measures to be used for comparison, such as NPV.

This page intentionally left blank.

Appendix G: Architecture Security Planning Considerations

Introduction

The purpose of this appendix is to describe the overall architecture security planning considerations that are an integral part of the standards-based planning process. It is essential to realize that IT security is not an add-on that can be fitted or not, like an optional extra for a car. IT security is both a mind-set and a management tool. It is not merely a concern for the confidentiality of data but also for its integrity and, most importantly, its availability.

IT security is not a negative, restrictive management tool but a facilitating one. Its purpose is to find a safe path through the hazards of business and technology problems. Two elements taken together form the purpose of IT security: the first is to ensure the availability of the resources of an organization to the potential user, when required, to the level required, and in safety; the second is to deny resource availability to unauthorized users. In essence, *IT security* equates with *resource maximization*.

The open systems/SBA concept represents a significant pattern or paradigm shift in the way in which 1) information technology is applied to data and information handling, and 2) the organization must be structured to make use of both.

Paradigm shifts have occurred in the past. The first occurred when organizations had to insert "data processing" into a completely manual organization. This produced the "fortress MIS" phenomenon. Security was relatively simple and, in most cases, merely required a wall be built around the mainframe computer.

The second paradigm shift was distributed systems when microcomputers spread like an infection to the extent that, in some organizations according to a recent report, there are more microcomputers than staff. This second shift presented a security problem in that it was no longer possible to put a wall around all the places where computing equipment appeared. Even IT planning became disseminated.

With the introduction of cooperative/networked processing on top of the unassimilated microcomputer spread, the pressure for change became so great that the degree of shift, or change in the paradigm, ushered in a new era in information handling. All that had gone before was referred to as Era I in DMR's *Strategies for Open Systems*, and all that followed the paradigm shift is Era II. This second era is one where the whole organization will be involved in information handling technology. If we were reliant on the computer before, we will be doubly so in the future. The organization will be planned around information flows and be fully dependent on IT technology. We will have come so far that it will be impossible for us to go back.

Under these circumstances, the applications that will be developed must be as reliable as possible while being flexible and responsive to change. This means that information protection requirements must be considered from the very beginning of IT planning, through to the stage where all the applications that are spawned are operational, and beyond.

The basis of the Era II environment is that a standards-based, networked infrastructure will become the norm, and that hardware, software, and applications can be "plugged in" easily. This has significant impact in terms of providing sufficient levels of security.

Security must be built into the infrastructure and into each feature using the infrastructure. The only effective way to do this is to insert security into the total IT process from architecture planning through to implementation.

If further justification is still needed for the use of IT security at all planning phases, consider that the thrust of the new SBA approach is to design for continuous change. Change means possible danger; if it is not monitored and controlled, a false step may lead to organizational damage and loss. The proposed control is through principles, generic models and the adoption of standards, and continual iteration. The result is a process that creates a systems environment that evolves and changes continuously rather that being cast in concrete. Under these circumstances, the widespread use of IT security is essential.

IT security architecture must produce the following for every application system or group of systems:

- A clear understanding of the security requirements and architecture for each application system and the IT security results of any interaction
- A detailed depiction of:
  - The IT security services and resulting mechanisms required
  - The boundaries of the IT security service
  - An overview, where possible, from beginning to end of the application or group of applications of the IT security service required.
- The capacity to apply different methodologies to the various application systems depending on and focusing on implementation requirements.

Planning the new architecture

Many organizations are now beginning to realize that they are all competing together in time. Their CEOs are demanding IT results now. The old static linear model, because it took too narrow a view of the business world, is now obsolete. Non-performance, or some form of extended response time, is no longer acceptable with the shorter planning cycle predicated by the new paradigm.

In such a speeded-up environment, it is easy to overlook the importance of IT security. In the push to get results, IT security and quality assurance are usually among the first things to be dropped or to which only lip service is paid. Business professionals who know what end results they want will often push for faster delivery times and deliberately overlook certain technical requirements for data and information protection. Their aim may be oversimplified as *"getting a working application as quickly as possible and with the minimum expenditure."* The technical specialists, on the other hand, are looking for the most efficient and effective technical solution. IT security can often be overlooked by both groups to the detriment of both their aims. Because of this, it is important to include on the AWG at least one IT security specialist who can identify the requirements rather than wait for a non-specialist to become familiar enough with the technology to be able to perform this service. It is easy for the unpracticed eye to overlook a situation that is a security situation in the making. Consideration of the five models and the architecture principles that lie at the heart of the standards-based architecture approach will show how intricately intertwined IT security is in the use of that approach.

Business model

This model identifies the business functions performed by the organization in fulfillment of its mandate. It also shows the informational flows required by each function and their interlinkages. This level is also the starting point for analyses of the impact on the organization of loss of each of the business functions. A business impact analysis of this type helps identify the levels of security required by each function. Coupled with an analysis of the recovery options, this will result in the development of contingency plans for the operation of each of those functions and for the organization as a whole. It can also be the starting point, depending on the criticality and size of the development effort, of a development contingency plan (see "Architecture Framework" below) designed to protect the development investment.

All through the planning process, the planning team must continually ask such questions as:

- Is this legal?
- Is this safe?
- What could go wrong?
- What are the risks attached to this decision and have they been evaluated?
- What is the level of risk involved in each case?
- What are the data protection, security, and safety aspects of the alternatives/proposed action?
- Which alternative is better from an IT security point of view?

Architecture principles

These act as the guides for the subsequent IT architecture views that will be developed. They should include the principles that begin to define the type of IT security or data protection architecture that the organization needs to support. To what level, for example, will subunits of the organization be allowed to handle their own IT security and how much, if any, central coordination will be provided? It is important to begin thinking of these things at the earliest possible stage. Protection and safety requirements can then be built in relatively easily, usually more cheaply, and certainly more effectively, than if they are retrofitted.

Work organization

This model provides an indication of the impact of the proposed changes on the organizational structure. Here the primary IT security concern is accountability. This is mainly a factor of responsibilities and their separation; for example, audit responsibilities should report to the highest level in the organization and should be independent of the line organization that must be audited. This avoids the situation where any individual or group is required to be judge and jury in its own case. The reporting responsibilities for security in general, and IT security in particular, are also important. Those positions responsible for granting access to the database, the issue and currency of passwords, and key management, for example, must be identified. There will, however, be other less obvious occurrences that must be identified and dealt with appropriately.

An important decision at this stage, if it has not already been mandated, is who is responsible for security. A number of legal decisions have been handed down in the United States where CEOs, whether they were aware of their responsibility or not, were fined and jailed for not adequately protecting their organization's data/information when "disasters" occurred. Consequently, if the decision is made that the user manages IT security with IT playing an advisory role, it is important to identify where the responsibility lies to ensure the user takes good advice, and who enforces it. If this is omitted, the lack of clear-cut responsibilities will usually result in time-wasting wrangling or a standoff in which nothing useful in the way of protection is achieved.

Information model

This model identifies the information requirements for the organization. For each data group identified, it must include the requirements for security as well as the data and information required by, for example, audit trails. Consideration must also be given to the advisability of mixing data and information of varying levels of sensitivity. Data aggregation can result in levels of sensitivity that the component data items do not attain.

Application model

This model analyzes and describes the functions and subfunctions that will be supported or automated through information technology and groups them into potential system applications. As part of this process, all logical dependencies and relationships among the application opportunity areas are identified. Defined at this stage are the scope and interfaces of applications that then provide the basis for detailed design. Identified at this time are IT security criteria that include:

- The sensitivity levels of the data handled by the various applications and the resulting sensitivity level of the applications
- The impact of linking applications of disparate sensitivities on potential users and on hardware and software choices
- The known security/protective strengths and weaknesses of the proposed hardware choices.

Technology model

The three components of the technology model define the hardware, software, and communications environment required to support the organization's business. Each element of these components requires an IT security profile showing not only its strengths and weaknesses but a general picture of what it can and cannot do and the way in which it does it. Thus, the hardware profile must include a definition of the security required to protect each element in conformity with the requirements identified for the business as a whole. This security profile, if not already identified, must be identified for each element considered by the planning team. The information derived from these profiles, if properly used, can improve the efficiency and effectiveness of the standards-based architecture being developed and will play a part in subsequent development decisions.

Implementation

There are seven phases in the planning process to implement a standards-based architecture in an organization:

1. Architecture framework
2. Baseline characterization
3. Target architecture

4. Opportunity identification
5. Migration options
6. Implementation planning
7. SBA administration.

The models discussed above fit into the target architecture phase and form part of the deliverable for that phase, the *Target Architecture Document*. However, all the statements made about the need to include IT security and information protection architecture considerations at the earliest possible point in the planning process still hold true. Consequently, elements of IT security will be found in each of the other six phases. Each phase is discussed in more detail below.

Architecture framework

This is a general definition of the current environment and the architecture direction to be taken for the target architecture. Any lapses in the current environment, as perceived by IT security, must be identified so that corrective action can be included in the new standards-based architecture. This means that a security review of the environment must be carried out for the organization or at least that area of it covered by the architecture being developed.

In developing the deliverable, the business and IT issues must be identified and the areas of interaction described in some detail. Where there is concern, for whatever reason, the causes must be outlined. Some problem areas will be apparent only as the result of identification by the security review, and some areas of general concern may have an IT security mandated solution.

The general description of the current IT organization, environment, and technology must include IT security, its responsibilities, and who is responsible for the delivery and enforcement processes included within it. There are a number of areas where IT security should operate, and its presence or absence should be noted; for example:

- Security administration roles and responsibilities
- Software development
- Change control

- Physical access controls
- Logical access controls
- Reliability and availability analyses
- Startup and shutdown procedures
- Security violation detection
- Protection from possible capture and/or overrun
- Key management
- Damage limitation
- Network management
- Recovery procedures and contingency planning.

These areas and others should be included the review of existing standards and any absences noted.

The review of existing opportunities should consider the impact of their implementation, from an IT security viewpoint as well as from others. At this point, all data elements to be handled by the standards-based architecture, which usually means the organization's total data holding, should have been reviewed and a sensitivity (confidentiality) level assigned. This, along with integrity and availability, determines the level of IT security required for the data covered by the architecture and the systems that handle that data. Without such determinants it is very difficult, for example, to be sure that the correct level of countermeasures has been applied. The cost of implementing the necessary data protection capabilities may vary significantly between the available opportunities. A wrong decision could result in significant additional, and unnecessary, costs in some instances. Security can be expensive, and money spent on protecting information assets that do not have a high value for one or more of the determinants may well be wasted. Also, a wrong decision at this point concerning opportunities could well alter a preference list based solely on other, non-IT security criteria.

Since IT security is really "good clean living with the computer," the architecture principles must include those that will protect the data and information in terms of the appropriate levels of confidentiality, integrity, and availability.

The other determinants of the level of IT security required in a system are:

- **Accountability:** This concerns the ability to identify and authenticate the source of an action and is essential to the audit process.
- **Access control:** This concerns the control of access to facilities and to components of systems. The controls may be mandatory (MAC) and rule based (RBAC), or discretionary (DAC) and identity based (IBAC). The controls may include labeling requirements and the restriction of downgrades and upgrades.
- **Non-repudiation:** In the transmission of data and information, it is important to know precisely who originated it and who received it. Therefore, proof of origin and proof of receipt are vital.
- **Assurances:** There must also be ways of assuring the users that the system architecture and the application planning and development process (systems development life cycle) can be relied upon to produce applications safe to use.

A potentially important consideration at this stage is the production of a development contingency plan. Depending on the size of the development effort and the criticality of the work being developed, a contingency plan should be put in place to ensure that the development work may be continued with the minimum of disruption and extra expense in the event of an emergency during the development period. As the development process continues, the cost increases. The loss of most or all of this development effort could be a severe setback to any development program because the replacement of the lost work may be impossible if the additional development funds are unavailable.

Baseline characterization

This activity defines the existing applications and technology platforms that form the foundation or baseline from which the standards-based architecture must develop. This definition phase includes a description of the baseline IT security measures in place for the protection of these existing applications and technology platforms.

Consequently, any imperfections in IT security terms and in the protective requirements of the baseline and the minimum level of IT security required across all the applications and platforms, must be identified. This may already have been done as the result of a security review or audit of some type. If it has not been done, then it must be done as part of this activity. Failure to do so runs the risk of building a new edifice (architecture) on faulty foundations. Deficiencies in the IT security baseline may then be made good before the new development begins or be planned as part of the new development work. Either way, the omissions will be remedied.

Target architecture

Using many of the directional elements developed in the architecture framework phase, this phase defines in greater detail the architecture aimed at or targeted. It should represent the *idealized vision of the architecture to be implemented* with the proviso that this idealized vision must make allowance for IT security requirements.

In developing a standards-based infrastructure architecture, the AWG takes all business, work organization, application, and information models as input, all of which have been considered from an IT security viewpoint. The target architecture phase uses those models to develop the architecture for the generic application and technology environments. In addition, the target standards and technology platforms on which those environments will reside are fully described. In this way, the IT security requirements are carried through what has been described as *"the essence of SBA planning."* Figure G-1 illustrates the familiar standards-based model, and every element indicated has an IT security aspect.

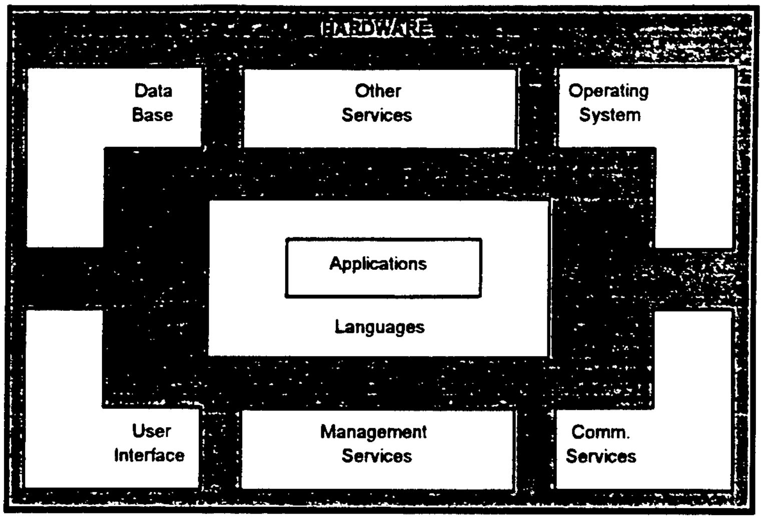


Figure G-1. Standards-Based Model

Opportunity identification

This phase takes a closer look at the opportunities identified in the previous phase, the target architecture. The opportunities identified may require researching and testing. This classifies them according to a number of criteria, including IT security criteria. In the case of software, the evaluation criteria, rationale, and guidelines for use are derived from DoD 5200.28.STD *Department of Defense Trusted Computer System Evaluation Criteria.* The IT security criteria for databases are provided in NCSC-TG-021 *Trusted Data Base Management System Interpretation Criteria.* These sets of criteria, depending on the circumstances, can have a significant effect on architecture flexibility and interoperability and, of course, on the costs.

Migration options

This phase involves sizing migration steps and identifying the "trigger points" on the implementation path where specific actions must take place for the successful implementation of the standards-based architecture. The migration path must allow for organizational change and must also be flexible enough to accommodate changes in the architecture itself that occur as the migration plan is being implemented. There are four areas where migration activity may be focused.

Work flow and organization

This includes the organization of work procedures and business operations at the user level and how users conduct business activities with regard to the active use of information technology. It is important to ensure that the organization of the work flow does not contravene any of the IT security principles, policies, and guidelines already identified. It is easy, when moving from generalities to the next level of detail down, to miss the observance of some security criterion agreed upon at an earlier stage.

Data and information

IT security must be sure that the data and information resources of the organization are not put to any uses that run contrary to IT security requirements and guidelines and do not contravene good management practice.

Applications

These are the tasks performed by IT or to which IT is *applied* in support of the business functions of the organizations. IT security must monitor a number of aspects of application development to ensure that reliable systems are produced to the correct level of security. Therefore, allowance must be made for IT security to perform such tasks as:

- The development process to ensure, for example, that no Trojan horses have been inserted in the code or security features disabled
- The quality assurance process
- The organization and level of separateness of the development, testing, operations, and maintenance units
- Applications handling data of disparate sensitivity levels are not linked.

Technology platforms

The underlying hardware, communications, and system software components used by the delivered applications have security strengths and weakness. IT security must ensure that the secure limits are not exceeded or liable to be exceeded. Thus, every effort must be made to avoid a security failure or incident.

Implementation planning

This phase, harvesting the benefits from the new architecture, endeavors to identify the short-term gains achieved. Once these have been identified, the focus becomes broadening the awareness of the successes throughout the organization to induce *"culture change."* An IT security awareness program should be considered as part of this. Part of the reason for the success of that particular project is improved availability and integrity measures built in as part of the application development process associated with standards-based open systems.

SBA administration

"Reality testing" the elements of the standards-based architecture once they have been implemented is done by conducting a comprehensive review of the *Architecture Framework Document* produced in Phase 1, as well as the *Baseline Characterization Document* produced in Phase 2 of the overall implementation process. The output is a self-critical document used to modify the overall *Architecture Framework Document.* This phase closes the loop in what is a cyclical process. Modifying the *Architecture Framework Document* starts the process afresh. IT security must, therefore, be represented in this phase, as in all others, to ensure that IT security requirements are not overlooked. They may be given due attention during the first iteration of the cycle but can subsequently be erased if they are not given additional attention.

New legislation or changes to old legislation may also require changes to the IT security infrastructure. Technological developments may necessitate changes or modifications to the IT security approach taken. Again, is flexibility is emphasized. Although IT security requirements must be considered and included at an early stage, they cannot be considered "set in concrete" or otherwise immutable.

The provision of IT security capabilities should not be seen as a hindrance to a project or as an unnecessary budget item. They identify an integral component of the information itself, and information handling in general—ease and safety of use. Our increasing reliance on computerized applications demands that this component be present so that effort can be concentrated on using information technology to the fullest, rather than worrying that the organization will be left high and dry by IT failure.

This page intentionally left blank.

Appendix H: How To Do SBA Administration

SBA administration

Most organizations recognize the "need for SBA governance" on or about the time that the initial SBA planning project comes to a close. It is strongly recommended that the DoD adopt a mechanism for keeping the SBA up to date.

It is not uncommon for an organization to establish an SBA administration function that coordinates the review of SBA-related projects and resets project priorities based on architecture evolution.

Typically, this coordination is managed through semi-annual SBA review meetings held with SBA representatives from each of the major functional areas participating in the SBA effort (representatives are selected by the ASC). SBA representative are responsible for keeping the SBA administration function abreast of changes in project status and direction. In turn, the SBA administration uses the representatives to execute changes in the general SBA strategy (consult the *Implementation Plan Document* for more details). The final pages of this SBA Guide describe a recommended process that can be used to support the goals of the SBA.

Process overview

An SBA Management Team (SBAMT) will be established to maintain the SBA. This team will work directly with project managers responsible for developing SBA projects as well as with the functional managers and their staff responsible for overseeing project implementation.

It is paramount that the SBAMT build into the overall administration process a review system to ensure compliance with the objectives set forth in the *Architecture Framework Document, Target Architecture Document, Opportunity Identification Document, Migration Options Document,* and *Implementation Plan Document.*

Monthly project coordination meetings will be held between the SBAMT and all project managers developing SBA-related efforts. The purpose of these reviews will be two-fold:

- Provide an opportunity for project managers to report any issues that will impact the delivery of their projects to the SBAMT, who will approve changes to project plans
- Create an environment whereby SBA project managers can meet to discuss cross-project issues and actively identify opportunities to reuse code and build integrated systems.

On a quarterly basis, the SBAMT will sponsor a status review with the executive sponsor. This quarterly review will provide top decision makers within the organization an opportunity to review the progress of key IT initiatives while lending guidance to the SBAMT.

When the SBAMT is not meeting with project managers or the executive sponsor, they are updating the SBA project plans and communicating all changes to these plans through a myriad of communication vehicles intended to provide needed information to all members of the organization's stakeholder community. (See the "communication vehicles" part of this appendix for more details.)

Key elements of the SBA management process

Following are several important elements in the SBAMT process:

- Establishment of the SBAMT
- Addition of SBA to the duties of the executive sponsor
- Implementation of the project coordination meetings
- Institutionalization of the quarterly SBA reviews.

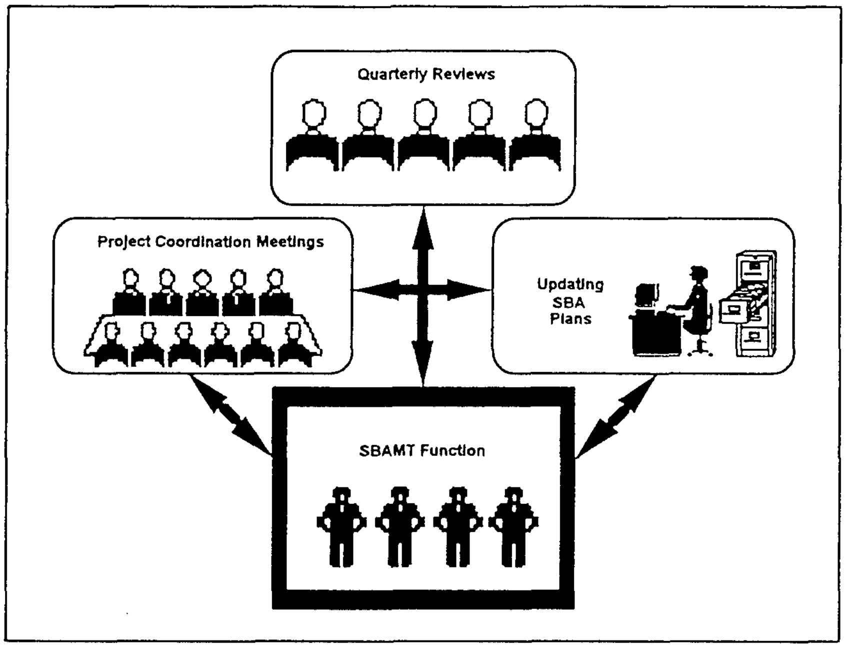


Figure H-1. The SBA Management Team (SBAMT)

The SBAMT

The first step in the SBA administration process is to establish an SBAMT. The SBAMT is charged with keeping the SBA up to date. This is done by managing the coordination of the projects defined in the SBA *Implementation Plan Document.* The people assigned to this function will employ such devices as monthly meetings with SBA project managers as well as quarterly reviews with the executive sponsor in order to ensure that the SBA projects are evolving as planned.

The team should be staffed with experienced planners and technologists who have a deep-rooted understanding of IT implementation projects (i.e., data processing, communications, and systems analysis). Typically, the team is situated in the IT systems development area enabling it to oversee the development activities. If not, standards and policies defined by the SBAMT could be ridiculed because the process "was not invented here." If reorganization occurs, it is important that the SBAMT be placed with the highest ranking IT officer to ensure continued execution of the SBA plans.

Many organizations are beginning to place SBA administration functions under the command of the senior-most executive (i.e., the CEO) in order to ensure that the most crucial IT applications are being developed in unison with the organization's strategic plan. This is highly recommended and represents the best case scenario.

Once established, the team must conduct a general assessment of the SBA projects to see if, in fact, the projects are being implemented in compliance with the overall architecture. This is done by mapping project progress against the implementation plans as well by as the team asking itself (and the responsible project managers) some hard questions like:

- Is the architecture framework still valid? Should any of the architecture principles be modified? Which ones and why? What has changed?
- What are the benefits to be had from changes to the implementation plans? Are there any cost savings, value-added benefits, or softer, long-term intangible benefits?
- Have IT standards been materially implemented in the organization? How far along the standards road have we traveled thus far? How far, given this "process check," do we have yet to go? Have we gleaned 80 percent of the benefit already, or is there still payoff down the road?
- Has the enterprise recognized any benefit from the work achieved?
- Given the current state of implementation, have any other payoffs been obtained that may not have been originally predicted?
- In general, do the plans and their delivery schedules appear to be changing?
- Have any standards, targeted as important, not yet matured as much as originally anticipated?
- What is the status of the technology that was selected for implementation? Has it "shown up on time" in the marketplace? Have we secured its acquisition?

After these questions have been answered, adjustments to the original plans should be made (i.e., if a given project is not maturing as originally scheduled, specific steps must be developed to produce "workarounds").

Primary responsibilities

- Conduct monthly project coordination meetings
- Conduct quarterly executive sponsor meetings
- Update SBA plans
- Communicate SBA changes to the stakeholder community
- Review SBA project status
- Facilitate cross-project sharing of information/code
- Identify opportunities to consolidate systems development efforts
- Assist project managers in adjusting SBA project plans
- Coordinate complimentary voice and data development efforts.

Executive sponsor

In industry, perhaps the largest constraint in SBA implementation work is senior management's unwillingness to participate in the review and nurturing of the IT architecture. To keep SBA in the forefront of activities in the systems development arena, this attitude must change.

An IT steering committee must be formed, charged with overseeing the prioritization of SBA projects, as well as final approval for all changes and adjustments to the SBA project scope and delivery schedules. This duty would be appropriate for the executive sponsor. This team of senior officers should be prepared to commit the necessary resources required to make SBA a success.

Typically, the steering committee (executive sponsor) members participate in quarterly reviews of the SBA project status and actively seek to incorporate input from the quarterly SBA reviews into their budget/planning (i.e., POM) process. These decision makers assist the SBAMT in implementing the necessary changes to the SBA by communicating shifts in priorities to their subordinates.

In this new kind of "top-down," "function-driven" environment, assessment and review become less personally and politically charged. The result is that the SBAMT process becomes easier to conduct successfully. Ultimately, this form of organizational behavior leads to the establishment of a successful and repeatable implementation process.

Primary responsibilities

- Participate in quarterly SBA reviews
- Make decisions regarding SBA project priorities and adjustments
- Oversee SBA project implementation within the functional areas of the enterprise.

Quarterly SBA reviews

Quarterly SBA reviews are a vehicle to help executive sponsor members keep abreast of SBA progress and be aware of all the changes that occur during the SBA project evolution. Information conveyed in these reviews should be incorporated into the budgeting process within the enterprise. In this way, the enterprise will reduce the dollars being squandered on insignificant IT projects.

Also, these reviews are an important means by which the SBAMT can gain an understanding of the desires of senior officers (i.e., balance current priorities with new requirements). This insight will be needed to better manage changes to the SBA project plans and to define new SBA projects.

Primary objectives

- Executive management review of the SBA progress
- Approval and prioritization of new SBA projects
- Approval and prioritization of changes to existing SBA plans
- Providing a means for functional areas to articulate new IT requirements.

Monthly project coordination meetings

Project coordination meetings are held between the SBAMT and all the SBA project managers responsible for building SBA projects. These meetings are a way for the administrators to understand the issues affecting SBA efforts, enabling them to make changes to the SBA.

Furthermore, these meetings are used to encourage project managers to discuss interproject issues, like software reuse and data integration. When this communication vehicle hits its stride, it can be used to deliver information regarding new IT standards and policies to all project managers represented in the coordination meetings.

Primary objectives

- SBAMT review of SBA projects (plans and budgets)
- Announcement of adjustments in SBA plans
- Cross-project discussions on coordination issues (i.e., data sharing, etc.)
- Delivery news on IT related issues (i.e., standards adoption, etc.).

Communication vehicles

As mentioned earlier, it is extremely important to staff the SBAMT with seasoned IT professionals. To do otherwise can be disastrous. Team members must come to the planning table with experience in technology planning and the sensibilities to understand the inherent cultural and political climate.

The next most important factor in conducting successful architecture administration is the establishment of a set of effective communication mechanisms that can help the administration team distribute important information, such as project planning documents, and receive critical feedback without having to become immersed in the typical "red tape" that such work usually entails.

Figure H-2 highlights this issue and suggests several ways the Marine Corps can keep the communication lines open while effectively distributing valuable information about the status of its SBA projects.

Quality review meetings

Sometime during the first year of SBA administration, the SBAMT should develop a quality review process that will be applied to each SBA project as it matures through the phases of the project development life cycle. This "process check" should conform to existing Total Quality Management (TQM) initiatives and, as such, provide a "quality assurance" dimension to the overall architecture administration process.

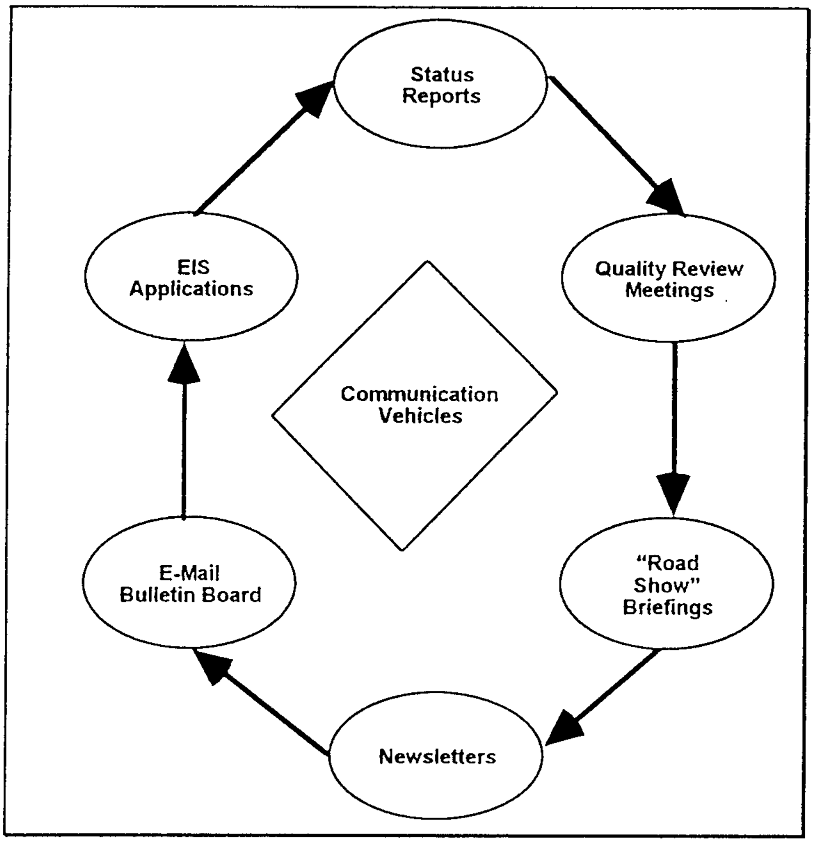


Figure H-2. Some Important Communication Vehicles

A review process based on the Continuous Process Improvement Cycle (see Figure 8-1) is recommended. The notion is that a project is planned, work begins, the result is checked against the plan, and opportunities for improvement are defined and acted upon through modifications to the next plan (or project phase, whatever the case may be). The use of this technique will help the enterprise learn from its SBA experiences.

Each review meeting can be used as a way for the SBAMT to communicate suggested changes in the project development process to SBA project managers (internal as well as external personnel), contributing to the creation of the "learning organization," which is fundamental to TQM objectives.

Status reports

Status reports are another way to improve communication within the SBA development environment. By documenting such things as causes of project delays or scope changes, the SBAMT can begin to define ways to proactively address them. These "lessons learned," together with the modified plans, should be included in a quarterly SBA status report and delivered to all designated personnel.

Often overlooked, documenting the "lessons learned" (see Figure 8-4) becomes very valuable to future project development teams, particularly when defining modifications to SBA project plans helping future project managers to "never make the same mistake twice."

"Road shows"

Another important way to inform enterprise personnel about the significance of SBA is to establish an SBA awareness program (or "road show"). The road show will involve the creation of an SBA briefing that describes the SBA process and explains the impact it has on the enterprise. (See Figure H-3.)

The SBAMT will schedule briefings at all major sites. All personnel would be expected to attend one of these briefings. Once all personnel have been exposed to the SBA project, the next phase of the awareness program would take the form of annual status meetings delivered at the same sites.

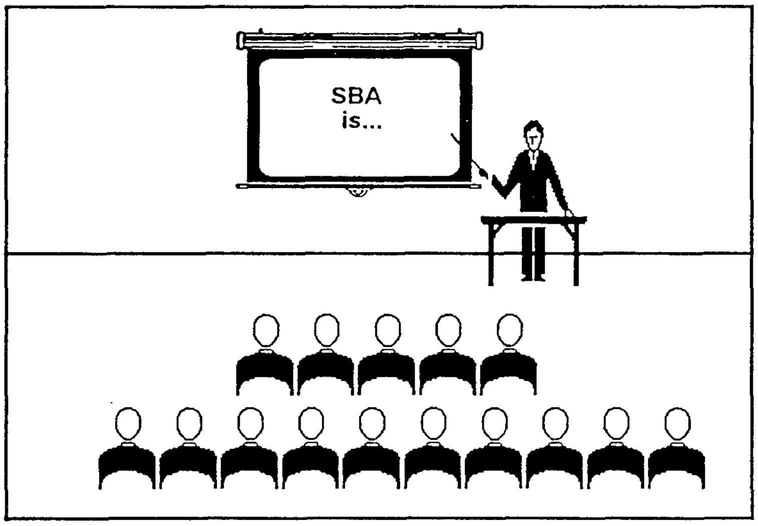


Figure H-3. The SBA "Road Show" Will Take the Message to the Troops

Newsletters

An SBA newsletter could also be created as a means of keeping all personnel informed of the SBA progress. The newsletter could be published quarterly, and its production should coincide with the IT executive sponsor meetings. This way, news concerning executive management decisions about SBA events can be delivered to the entire community.

Electronic bulletin boards

An electronic bulletin board dealing with SBA subjects can be established within the E-mail environment. (See Figure H-4.) It can become a very useful broadcast mechanism, since many personnel use it on a daily basis. In fact, many organizations in the commercial world use such devices as a way to solicit improvement ideas from personnel, transmit newsletters, distribute results from quarterly reviews, and deliver project progress reports to SBAMT-like groups.

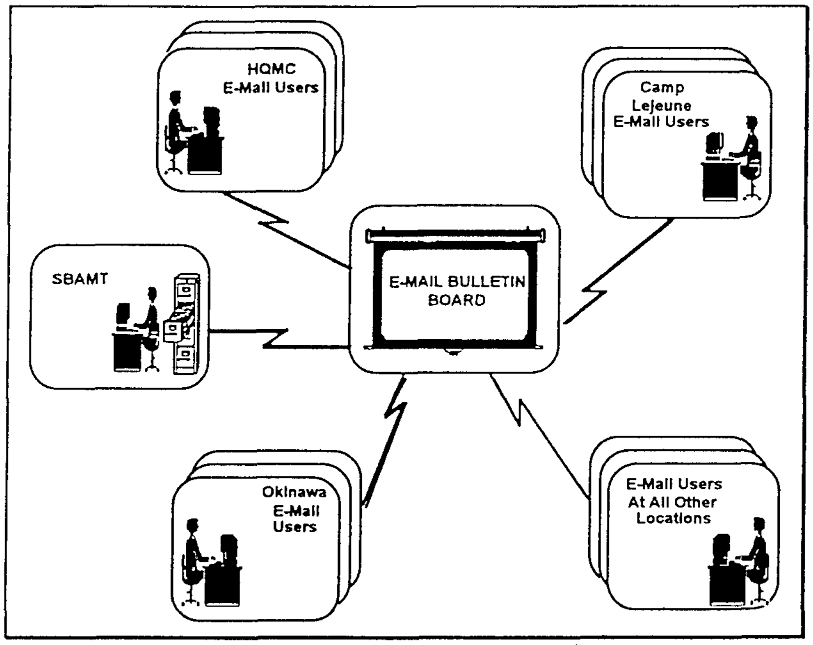


Figure H-4. The E-mail Bulletin Board Posts All SBA News for All Personnel to Access

EIS applications

The development of an SBA Executive Information System (EIS) is another effective communication tool. The primary focus of such a system is to provide an electronic means of keeping senior management aware of changes in SBA projects.

The typical EIS system is easy to use, has user-defined triggers and a myriad of other features that make such a system a very useful tool. (For example, each executive can define areas of particular interest so that when one of his SBA projects is affected in any way, an electronic message is sent to his computer; similarly, other changes that are not of interest never show up on his screen).

Architecture remodeling

When should you remodel? When any of the principles developed in the architecture framework phase have changed. Another reason could be a major change in technology significant enough not to have been anticipated in the target architecture phase; however, such changes will become increasingly rare. One of the major benefits of standards planning is that standards, unlike the underlying technology itself, change far more slowly.

In theory, one should never have to change the architecture if the architecture principles do not change; however, they do change from time to time. When this happens, the SBAMT should discuss and confirm the perceived changes with the SBA executive sponsor and all IT project managers before taking any action.

This page intentionally left blank.

Appendix I: Sample Deliverable Table of Contents

This section provides general outlines for each of the deliverables in the SBA planning process. These may be amended and customized by the AWG for presentation to the ASC. The individual circumstances surrounding the organizational culture and IT environment will also influence the deliverable.

The standards-based architecture

The standards-based architecture is composed of seven deliverables, which are released on a phased basis. Figure I-1 outlines the individual components of the model.

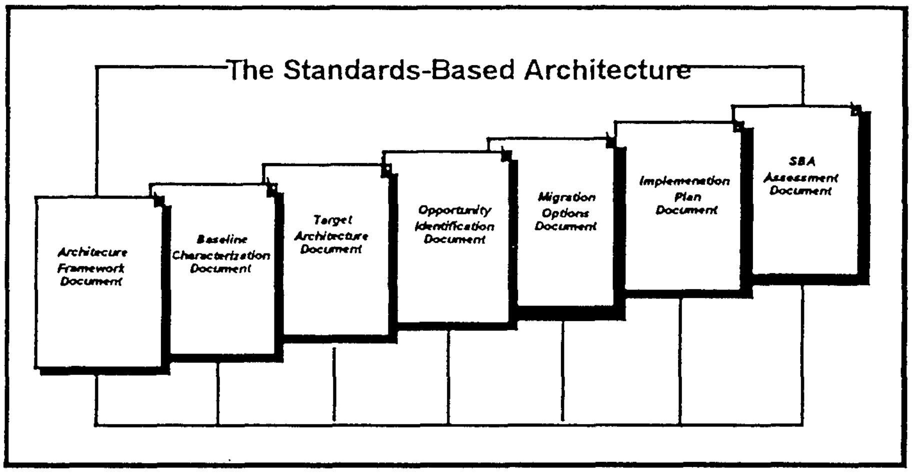

The Standards-Based Architecture
Architecure Framework Document
Baseline Characterization Document
Target Architecture Document
Opportunity Identification Document
Migration Options Document
Implemenation Plan Document
SBA Assessment Document

Figure I-1. The Standards-Based Deliverable Set

Staged deliverables throughout the process

A key aspect of the standards-based planning process is the manner in which the architecture is developed. It is recommended that at each phase of the planning process an interim deliverable be produced by the team. Figure I-2 illustrates the phases and their associated deliverables.

Deliverable style

All of the deliverables should be "executive style" in scope, easy to read, and highly visual in nature. The key attribute of these deliverables is that they are distributed across the organization and are used to communicate the chief attributes of the architecture to the various constituencies within the enterprise.

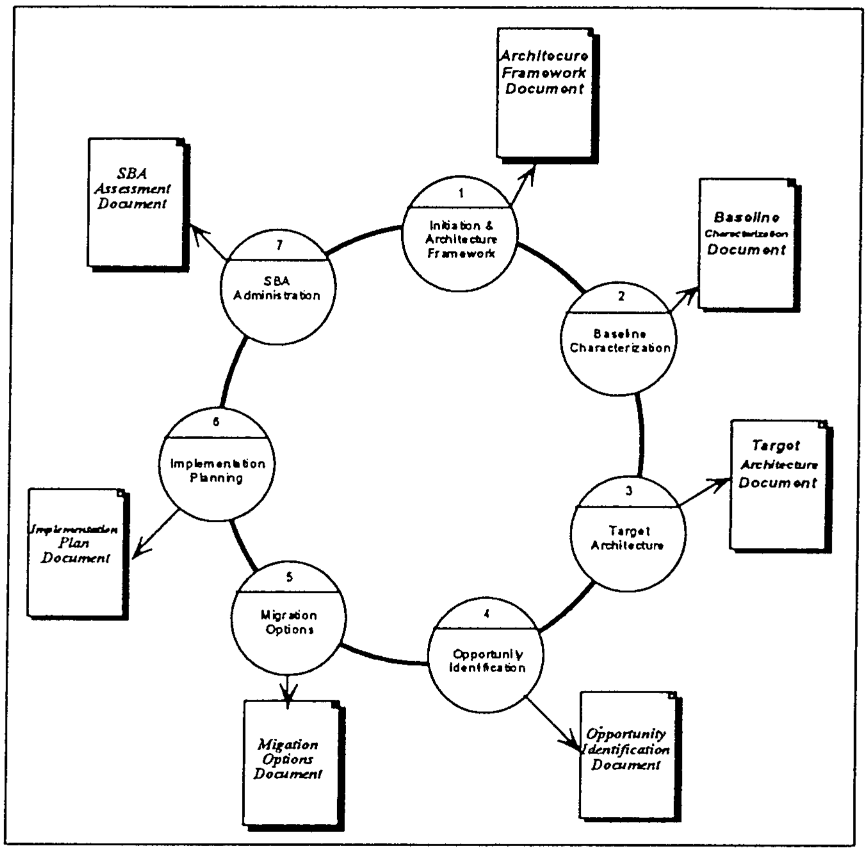

Architecure Framework Document
SBA Assessment Document
1
Initiation & Architecture Framework
Baseline Characterization Document
7
SBA Administration
2
Baseline Characterization
6
Implementation Planning
Target Architecture Document
3
Target Architecture
Implementation Plan Document
5
Migration Options
4
Opportunity Identification
Migation Options Document
Opportunity Identification Document

Figure I-2. The Standards-Based Deliverable Set

The length of each document should be between 25 and 45 pages. This will assure that the documents actually get read by individuals in the organizations.

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Key functional drivers and issues

III. Key interview findings

IV. IT principles constitution

V. Architecture planning issues

- Functional technology issues
- IT description: current environment
- Security issues
- Cost/benefit design concerns

VI. Functional and information opportunities

VII. Design issues

- Design principles, guidelines, and implications
- Design alternatives review
- SBA design attributes

VIII. Next steps

Baseline Characterization Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Key architecture baseline characterization issues

III. Scope and approach

IV. Classification and description

- Platform classification
- Generic application model
- Generic technology model
- Work flow model
- Generic information model
- Standards support description
- Security evaluation
- Connectivity support model
- Cost/performance data

V. Summary assessment of design issues and constraints of current environment

VI. Implications for target architecture design

VII. Next steps

Target Architecture Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Target architecture description

- Work flow and processes
- Data and information
- Applications
- Technology platforms
- Standards
- Migration issues
- Architecture organization and personnel issues

III. Architecture design alternatives

IV. Procurement issues

V. Implementation issues

VI. Next steps

Opportunity Identification Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. Implementation opportunity identification

- Strategic opportunities
- Major opportunities
- Quick hits
- General benefit and business case
- Magnitude, payoff, and degrees of freedom classification

III. Overall benefit classification

IV. Next steps

Migration Options Document

SAMPLE TABLE OF CONTENTS

I. Executive summary

- Project status
- Key issues

II. General cost/benefit definition

III. Migration project scope definition

IV. Technology standard implementation strategy

V. Time lines and trigger points

VI. Project cost and time frame considerations

VII. Specific business case and cost/benefit analysis for identified opportunities

VIII. Project deliverables definition

IX. Organizational change process requirements

X. Next steps

Implementation Plan Document(s)

This is not a formal presentation document, rather it is the aggregate set of project plan documents produced by the individual functional unit.

Presented below is a suggested set of topic areas to include in each plan. These may vary widely depending upon the implementation project but should comply with all DoD project management standards.

I. Project description

II. Objectives

III. Scope

IV. Deliverables

V. Critical success factors

VI. Constraints

VII. Task list

VIII. Effectiveness measures

IX. Technology requirements

X. Staffing skills

XI. Completion criteria

XII. Other issues

SBA Assessment Document

SAMPLE TABLE OF CONTENTS

I. Executive summary
- Project status
- Key issues

II. Scope of architecture review

III. Key review findings

IV. Implementation adherence to IT principles and target architecture
- Processes
- Information
- Platforms
- Standards
- Migration issues
- Architecture organization and personnel issues

V. User views of benefits and functionality delivered

VI. Review of cost/benefit implementation delivered

VII. Continuous process improvement recommendations

VIII. Next steps

This page intentionally left blank.

Appendix J: Glossary

American National Standards Institute (ANSI): The principal standards coordination body in the United States. ANSI is a member of the International Organization for Standardization (ISO).

Application: The use of capabilities (services and facilities) provided by an information system specific to the satisfaction of a set of user requirements. [P1003.0/D15]

Application Entity: The part of an application process that interacts with another application process.

Application Layer: Layer seven of the OSI Reference Model. It serves as a window through which applications access communication services.

Application Model: A term used to describe those functions of an organization that can be supported or automated through IT. It is used for grouping or clustering functions into applications. It provides the application developers' views of the IT architecture.

Application Process: The part of an application that resides in a single end system.

Architecture: Architecture has various meanings depending upon its contextual usage.
(1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE STD 610.12]
(2) Organizational structure of a system or component. [IEEE STD 610.12]
(3)The disciplined definition of the IT infrastructure required by a business to attain its objectives and achieve a business vision. It is the structure given to information, applications, and organizational and technological means—the groupings of components, their interrelationships, the principles and guidelines governing their design, and their evolution over time.

Bridge: The hardware and software used to connect circuits and equipment in two networks with the same protocol.

Common Applications Environment (CAE): The X/Open term for a computer environment in which applications can be ported across X/Open vendor systems. It includes standards for the operating system, languages, networking protocols, and data management.

Computer-Aided Acquisition and Logistics Support (CALS): Standards for electronic file format interchange and data management adopted by the U.S. Department of Defense to acquire, process, and disseminate technical information in digital form. CALS will facilitate the transfer of logistic and technical information between industry and Government by leveraging existing international standards. Among the industry standards used in CALS are IGES (CAD, vector graphics), SGML (automated publishing), GRP 4 Raster or TRIF (raster scanned images), and CGM (illustrations).

Computer-Aided Software Engineering (CASE): A set of software tools that automate and contribute to the improvement of the software development process.

Conformance: Meeting standards. By running standard test scripts, conformance testing ensures that a product meets standards.

Connection: In data communications terminology, a logical link established between application processes that enables them to exchange information. In the OSI Reference Model, an association established by one layer with two or more entities of the next higher layer for the transfer of data. In TCP/IP, it is a logical TCP communication path identified by a pair of sockets, one for each side of the path.

Data Link: An assembly of two or more terminal installations and an interconnecting line.

Data Link Layer: Layer two of the OSI Reference Model. It controls the transfer of information between nodes over the physical layer.

Directory Services: A service of the External Environment entity of the Technical Reference Model that provides locator services that are restricted to finding the location of a service, location of data, or translation of a common name into a network specific address. It is analogous to telephone books and supports distributed directory implementations. [TA]

Distributed System: A system consisting of a group of connected, cooperating computers.

Distribution List: A list containing the names of mail users and/or other distribution lists. It is used to send the same message to multiple mail users. It can be private or public.

Electronic Mail: The electronic generation, transmission, and display of correspondence and documents. Electronic mail is a GAE.

Entity: An active element within an open system layer (e.g., session entity, transport entity). It can represent one layer, one part of a layer, or several layers of the OSI Reference Model. One layer can include several entities.

Exterior Gateway Protocol (EGP): The service by which gateways exchange information about what systems they can reach.

Gateway: A device for converting one network's message protocol to the format used by another network's protocol. It can be implemented in hardware or software.

Generic Application Environment (GAE): A term used to describe the set of architecture components that describe the different possible types of IT applications.

Generic Technology Environment (GTE): A term used to describe the set of architecture components that describe the different types of services required to support a GAE.

Generic Technology Platform (GTP): A term used to describe the different types of delivery components that can be used to support IT applications.

Government Open Systems Interconnection Profile (GOSIP): A government (e.g., U.S. or U.K.) profile of functional applications that outlines a national policy and strategy for converting to a communications system based on OSI. Use of GOSIP is no longer mandatory.

Host: A computer, particularly a source or destination of messages, on a communications network.

Information Model: A term used to describe the information resources of the organization and their interrela-tionships. It is used to support data modeling and resulting database and document storage design requirements. It provides the information resource managers' views of the architecture.

Institute of Electrical and Electronics Engineers (IEEE): An accredited standards body that has produced standards such as the network-oriented 802 protocols and POSIX. Members represent an international cross section of users, vendors, and engineering professionals.

Integrated Services Digital Network (ISDN): The recommendation published by CCITT for private or public digital telephone networks where binary data, such as graphics and digitized voice, travel over the same lines. ISDN will unite voice and data transmission, including imaging, over the same kind of digital network that links most telephone transmissions in use today.

Interface: A connecting link between two systems. In the OSI Reference Model, it is the boundary between adjacent layers.

International Standard (IS): Agreed international standard as voted by ISO.

International Organization for Standardization (ISO): An organization that establishes international standards for computer network architecture. Its OSI Reference Model divides network functions into seven layers. (Membership is by country, with more than 90 countries currently participating.)

Interoperability: (1) The ability of two or more systems or components to exchange and use information. [IEEE STD 610.12]. (2) The ability of the systems, units, or forces to provide and receive services from other systems, units, or forces, and to use the services so interchanged to enable them to operate effectively together. The conditions achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. [Joint Pub 1-02, DoD/NATO] [JOPES ROC]

(2)The ability of applications and computers from different vendors and architectures to work together on a network.

Interoperability Testing: Procedures for ensuring that a computer product or system can communicate in a multivendor network.

Layer: A level of the OSI Reference Model. The model divides functions for transferring information between systems into seven layers, grouping the related functions or tasks and making them easier to understand. Each layer performs certain tasks to move the information from sender to receiver. Protocols within the layers define the tasks for networks but not how the software accomplishes the tasks. Interfaces pass information between the layers they connect.

Local Area Network (LAN): A data network, located on a user's premises, within a limited geographic region. Communication within a local area network is not subject to external regulation; however, communication across the network boundary may be subject to some form of regulation. [FIPS PUB 11-3]

Message: A block of information sent from a source to one or more destinations.

MS-DOS: The personal computer operating system developed by Microsoft Corporation.

Multivendor Network: A computer network with hardware and software from more than one vendor.

National Institute for Standards and Technology (NIST): The division of the U.S. Department of Commerce that ensures standardization within Government agencies. NIST is responsible for the Applications Portability Profile—a set of standards and guidelines for U.S. Government procurement. NIST was formerly known as the National Bureau of Standards (NBS).

Network: A system of connected computers.

Network Layer: The third layer of the OSI Reference Model. This layer controls underlying telecommunication functions such as routing, relaying, and data link connections.

Node: A point in a network, either at the end of a communication line (end node) or where two lines meet (intermediate node).

Open Network: A network that can communicate with any system component (peripherals, computers, or other networks) implemented to the international standard (without special protocol conversions, such as gateways).

Open Software Foundation (OSF): An organization created by major IT vendors to define specifications, develop software, and make available an open, portable environment.

Open Systems: (1) A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [P1003.0/D15] (2) Software environments consisting of products and technologies that are designed and implemented in accordance with "standards" (established and de facto) that are vendor independent and commonly available.

Open Systems Interconnection (OSI): A set of standards that, when implemented, let different computer systems communicate with each other.

Operating System: A group of programs operating under the control of a data processing monitor program. It manages such functions as memory, processing tasks, and interprocess communication in a computer system.

OSI Reference Model: The seven-layer model, defined by the ISO, that provides the framework for building an open network. The seven layers, ranging from highest to lowest, are application, presentation, session, transport, network, data link, and physical.

Password: A string of characters required to gain access to directories, files, or applications.

Peer Protocol: The protocol governing communications between program entities that have the same function in the same layer in each of two OSI networks.

Physical Layer: The first layer of the OSI Reference Model. It governs hardware connectors and byte-stream encoding for transmission. It is the only layer that involves a physical transfer of information between network nodes.

Portable Operating System Interface for Computer Environments (POSIX): An IEEE standard operating-system interface defining the external characteristics and facilities required to achieve the portability of applications at the source-code level.

Portability: (1) The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE STD 610.12] (2) A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment. [IEEE TUTOR] (3) The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TA]

Porting: The process by which a software application is made operational on a computer architecture different from the one on which it was originally created.

Presentation Layer: The sixth layer of the OSI Reference Model. It allows an application to properly interpret the data being transferred.

Process: A general term for any computer operation on data.

Profile: A set of one or more base standards, and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [P1003.0/D15]

Protocol: A set of rules governing network functionality. The OSI Reference Model uses sets of communication protocols to facilitate communication between computer networks and their components.

Quality of Service (QOS): A set of characteristics of a connection as observed between the connection end points. In the OSI session and transport layers, acceptable QOS values are negotiated between the service users when the connection is established.

Scalability: The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII] The capability to grow to accommodate increased work loads.

Server Type: A class of servers in a client/server architecture.

Service Provider: The resource that provides the facilities of the relevant OSI Reference Model layer. The OSI session and transport layers are the service providers for the session and transport services, and the X.25 network gateway or X.25 message control system is the service provider for the network service.

Service User: The software application using the facilities of one of the layers of the OSI Reference Model. For example, a program that calls the programmatic interface to the session layer is a session service user.

Session Layer: The sixth layer of the OSI Reference Model. It provides the means for two session service users to organize and synchronize their dialogues and manage the exchange of data.

Store-and-Forward Message System: The communication process that allows messages to be stored at intermediate nodes before being forwarded to their destination. X.400 defines a message handling system that uses this process.

System:–People, machines, and methods organized to accomplish a set of specific functions. [FIPS PUB 11-3]

TCP/IP Gateway: A device, or pair of devices, that interconnects two or more networks or subnetworks, enabling the passage of data from one (sub)network to another. In this architecture, a gateway contains an IP module and, for each connected subnetwork, a subnetwork protocol (SNP) module. The routing protocol is used to coordinate with other gateways. A gateway is often called an IP router.

Technology Model: A term used to define and describe the components of the technology infrastructure that support the other architecture models. It is in this area that the enabling effect of standards-based architectures is felt the most. The technology model provides the technology managers' views of the architecture.

UniForum: A trade association dedicated to promoting UNIX and open systems. UniForum sponsors UNIX events, publishes magazines, directories and technical overviews, and proposes specifications.

UNIX: An operating system that has become a de facto industry standard, supported on a wide range of hardware systems from a variety of vendors.

UNIX International: The consortium that defines and promotes the UNIX operating system and related software products.

Wide-Area Network (WAN): A public or private computer network serving a wide geographic area.

Work Organization Model: A term used to describe the impact on business operations at the work group and user

levels. It is used by organizational change designers to manage the impact of introducing new IT systems. It provides the users' views of the architecture.

X.25: Recommendations developed by CCITT that define a protocol for communication between packet-switched public data networks and user devices in the packet-switched mode.

X.400: The international standard for a store-and-forward message handling system in a multivendor environment.

X/Open Company Ltd.: *A* nonprofit corporation made up of vendors and large corporate users who are investing in the specification of the X/Open Portability Guide (XPG), an open environment based on standards. X/Open also brands products.

This page intentionally left blank.

Appendix K: Proposing Changes to TAFIM Volumes

Introduction

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown below. Guidance on the use of the format is subsequently provided.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown below, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail:
TAFIM
Booz•Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/671-7937**; indicate "TAFIM" on cover sheet.

TAFIM Change Proposal Submission Format

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

Format Guidance

The format should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**a**) of the format would identify that individual. The information in the **Document Identification** part of the format (**b**) is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts (**c**, **d**, or **n**) of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: 'The working baseline will only be available to the TAFIM project staff.'" The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The c (5), d (5), or n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in c (6), d (6), or n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 5: Program Manager's Guide for Open Systems

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

This is the first release of Volume 5, *Program Manager's Guide for Open Systems*. This document release is intended to generate comments and feedback from the Department of Defense (DoD) information management (IM) community.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF EXPECTED UPDATES

Volume 5 is still a prototype document in many respects. Authors and subject matter experts are currently reworking several sections to address both user comments and previously identified needs. Sections of the document remain incomplete due to the unavailability of information and/or time and funding. Volume 5 will, however, continue to evolve and be adjusted to reflect the IM community's need for program management guidance.

In addition to harmonization with the documents listed above, the next version of Volume 5 will reflect:

- The results of interviews currently being conducted with DoD C4I and information systems program managers

- Review comments and feedback on this version of the document received from the IM community

- The coordinated definitions being developed by DISA/D5 in the draft document *Information Systems Architecture Relationships and Definitions* that is being staffed separately.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group (AMWG) will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.

- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.

- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the Defense Information Systems Agency (DISA) Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. DISA is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Other working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

CONTENTS

APPENDICES

FIGURES

This page intentionally left blank.

1.0 INTRODUCTION

1.1 PURPOSE

The purpose of this volume of the Technical Architecture Framework for Information Management (TAFIM) is to provide program managers and their supporting Government and contractor staffs with guidance for developing technical architectures in planning and managing command, control, communications, computers, and intelligence (C4I), and information systems programs, either migration or new acquisition programs. Volume 5 is a guide for applying and integrating the principles and guidelines of the TAFIM and other Department of Defense (DoD) guidance documents promoting an open systems environment (OSE) for information systems. The information provided in this volume is intended to assist C4I and information systems program managers in making sound management decisions that result in OSE-compliant systems.

1.2 SCOPE

Volume 5 contains guidance for those C4I and information systems program management areas where OSE principles and standards should be incorporated in planning and management. This guidance applies to all DoD Components in the management of new C4I and information systems, the modernization of existing C4I and information systems, and the upgrade of existing C4I and information systems components under the direction of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD/C3I). This includes all C4I and information systems programs, projects, activities, and information systems (including migration systems) that are to be acquired and managed in accordance with the DoD 8000 series directives and are subject to the TAFIM.

Volume 5 is currently in its first version; however, it encompasses and supports the information contained in the most recent issues of the other TAFIM volumes. As the TAFIM and new and existing C4I and information systems policies and directives emerge and evolve, Volume 5, following the approval and publication of this version, will also evolve to reflect the latest guidelines and resources available.

1.2.1 Intended Audiences and Uses

Volume 5 has several intended audiences. The primary audience consists of the chartered C4I and information systems program managers within the DoD Components. Additional audiences comprise other DoD C4I and information systems managers and their staffs, to include support contractors, involved in TAFIM-related activities. The use of Volume 5 is essentially the same for all audiences — to provide insight into the TAFIM and help locate required information concerning a variety of functional and technical topics related to C4I and information systems architectures and OSE. The volume also points to the other TAFIM volumes and additional

DoD information sources that will provide more in-depth explanation and assistance on a selected subject area. All publications cited as references can be found in Appendix C.

1.3 BACKGROUND

An information system includes support and mission-oriented applications, computing platforms, and communications networks. The current DoD information system technical infrastructure consists largely of stovepipe, single-purpose, and inflexible systems that are costly to maintain. These systems reflect a multiplicity of approaches to migrate toward open systems, with each system progressing along its own path with limited attention to interoperability.

The evolving DoD enterprise vision for information management (IM) emphasizes integration, interoperability, flexibility, and efficiency through the development of a common, multipurpose, standards-based technical infrastructure. This vision requires a new paradigm for building technical architectures and information systems that improve the effectiveness of functional operations and promote efficient use of technology throughout the DoD. In support of the DoD IM vision and goal, the TAFIM provides the single DoD technical architecture framework for managing multiple technical architecture initiatives and also provides the prescribed guidance and basis for evolving the DoD's technical architecture toward the DoD OSE initiative. Its use is directed in the series of DoD memoranda identified in Section 1.4 that mandate the TAFIM for this purpose.

The TAFIM consists of a cornerstone set of documents, including this document, which provide sound guidance for ensuring improved user productivity, development efficiency, portability, scalability, interoperability, and system security, while promoting vendor independence and reduced life-cycle costs. Currently, the TAFIM includes the following eight volumes:

- **Volume 1 - Overview.** Provides an overview of the TAFIM.

- **Volume 2 - Technical Reference Model (TRM).** Provides the conceptual model for information services and their interfaces.

- **Volume 3 - Architecture Concepts and Design Guidance.** Provides concepts and guidance to support the development of technical architectures.

- **Volume 4 - DoD Standards-Based Architecture Planning Guide.** Provides a standards-based architecture planning methodology.

- **Volume 5 - Program Manager's Guide for Open Systems.** Provides guidance to ensure that the principles and objectives of open systems are used in developing technical architectures and in planning and managing C4I and information systems programs.

- **Volume 6 - DoD Goal Security Architecture.** Addresses security requirements commonly found within DoD organizations' missions.

- **Volume 7 - Adopted Information Technology Standards (AITS).** Provides the DoD profile of standards and guidance in terms of TRM services and interfaces.

- **Volume 8 - Human Computer Interface (HCI) Style Guide.** Provides a common framework for HCI design and implementation.

The TAFIM embodies effective, flexible interoperability and integration capabilities and helps identify and establish a uniform and cohesive architecture framework and guidance structure for the establishment of technical architectures. While the TAFIM does not provide a specific architecture, the intent is to provide the assistance, services, standards, design concepts, and configuration that can be used to guide the development of technical architectures that meet specific mission requirements. It is independent of mission-specific applications and their associated data and can be applied to all information systems technical architectures, in all DoD organizations and environments (e.g., strategic, tactical, sustaining base).

As a whole or by independent volume, the TAFIM is a valuable tool for program managers in carrying out their information technology (IT) duties and responsibilities. To assist program managers in utilizing the TAFIM and meeting its objectives, TAFIM Volume 5 has been prepared to provide guidance in those program management areas where the incorporation of TAFIM principles and guidelines will assist in meeting DoD OSE objectives.

1.4 DOD POLICY ON TAFIM APPLICATION

The following DoD memoranda mandate the TAFIM as DoD-wide, IM technical architecture guidance and address its use in systems migration, data standardization, and process improvement:

- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum, "Technical Architecture Framework for Information Management (TAFIM)," 30 March 1995.

- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum, "Selection of Migration System," 12 November 1993.

- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum (with attachment), "Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement," 13 October 1993.

Appendix D contains the text of these and other pertinent policy documents addressing the use of the TAFIM.

1.5 PROPOSING CHANGES TO TAFIM DOCUMENTS

Appendix G contains the guidance and directions for submitting a proposed change to the TAFIM, including this Volume 5.

1.6 DOCUMENT OVERVIEW

Volume 5 contains four sections and nine appendices, as described in the following table.

Section	Description
1 Introduction	In addition to this document overview, Section 1 contains the purpose and scope of Volume 5; the background and purpose of the TAFIM, including relationship of Volume 5 to the other TAFIM volumes; DoD policy mandating the use of the TAFIM; and information on proposing changes to TAFIM documents.
2 Overview of Open Systems Architecture Objectives	Provides the definition of OSE and addresses OSE in relation to the evolution of the current DoD technical infrastructure and its guiding principles.
3 Areas of OSE Concern in C4I and Information Systems Program Management	Describes and addresses those elements of program management where OSE principles and standards should be incorporated into the C4I and information systems management process.
Appendix A: Acronyms	Contains a list of acronyms.
Appendix B: Definitions	Provides definitions of the terms used in Volume 5.
Appendix C: References	Contains a table of all resource documents cited in Volume 5 and their sources.
Appendix D: TAFIM Policy Memoranda	Contains the text of all policy memoranda pertaining to the TAFIM.
Appendix E: Systems Engineering Elements/Activities and Products	Contains a table describing the various elements and/or activities of Systems Engineering process discussed in Section 3.15.
Appendix F: DISA OSE Information Services	Contains a table of services available from DISA that can provide support to activities using the TAFIM.

Section	Description
Appendix G: Program Management Responsibilities Matrix	Contains a matrix of all program management activities discussed in Volume 5; the documentation to be produced in relation to each activity; and the DoD management level(s) responsible for the activities and products identified.
Appendix H: Proposing Changes to TAFIM Documents	Contains instructions for submitting TAFIM changes.
Appendix I: Information System Architecture Relationships and Definitions	Contains a definitive set of architecture components and definitions to structure the complexity of architecture related phrases used within the DoD.

This page intentionally left blank.

2.0 OVERVIEW OF OPEN SYSTEMS ARCHITECTURE OBJECTIVES

This section provides the definition of OSE and its purpose in the evolution of the current DoD technical infrastructure. The guiding principles or characteristics of an open system are also discussed in relation to their role in the design and development of OSE-compliant systems.

2.1 EVOLUTION TO OPEN SYSTEMS

The DoD technical infrastructure is evolving into an open system environment in response to a real need for information and resource sharing across differing or incompatible levels of information ownership (i.e., enterprise). As computer technology evolves, so do the practices and methodologies employed to integrate new technologies into the workplace. Included are the many principles developed for software engineering, which continue to be expanded upon and enhanced to guide/define the open systems environment.

Computer programming has evolved into software engineering in large part because of emerging requirements for software interfacing, structured programming, data sharing, distributed environments, etc. These requirements in turn have resulted in the introduction/acceptance of shared databases, relational database management systems (DBMSs), modularization (functional separation), software reuse, data standardization, standard interfaces, and the development of American National Standards Institute (ANSI), International Organization for Standardization (ISO), and Institute of Electrical and Electronics Engineers (IEEE) standards. As these requirements and practices have been applied at the system level (i.e., within a system), their intrinsic value has been recognized as applicable at the functional level (i.e., between systems). Figure 2-1 shows the relationships of systems within a functional area (arrows indicate information flow). As systems proliferate, the need for inter-system communications/integration at the functional level becomes clear. As technology advances, it becomes more and more important that each system be able to "talk" to other systems, within and outside of its own functional area. With these new requirements comes the further development of interface standards, refinement of data standards, categorization and allocation of services, etc. With the advent of networks and the introduction of open systems, more effective communication has become possible within and across functional areas, as depicted in Figure 2-2 (arrows indicate communication flow), as well as between the various levels of the Enterprise Model described in TAFIM Volume 1, Section 5.

The DoD IM Integration Model, also depicted in TAFIM Volume 1, Section 5 (Figure 5-1) shows the various interfaces across the Enterprise Model. As these possibilities for communications have emerged, so has the need for a DoD-wide open information infrastructure to support the various Services and missions of the defense community. In response to this need, the concept of the Defense Information Infrastructure (DII) has been developed.

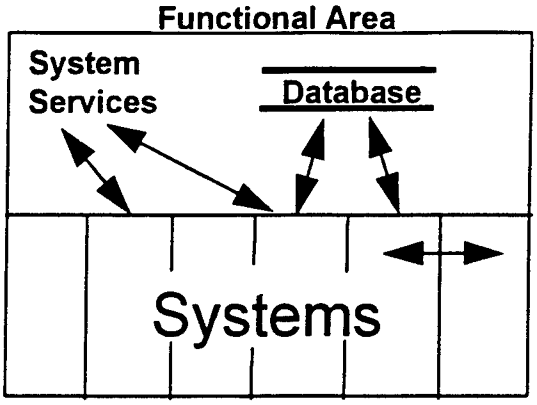


Figure 2-1. System Interfaces

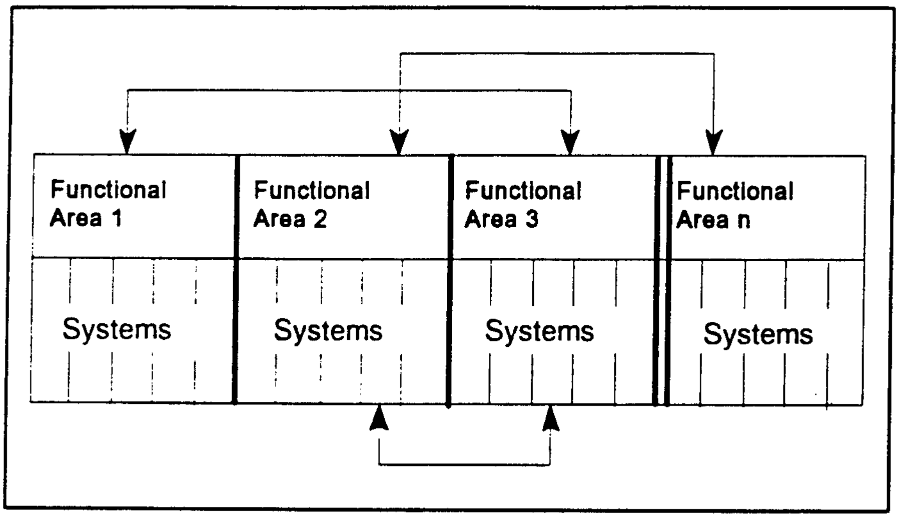


Figure 2-2. Functional Interfaces

The DII is envisioned to be a "...seamless web of communications networks, computers, software, databases, applications, data, and other capabilities that meets the information processing and transport needs of DoD users..."[1]

The goal architecture of the DII includes the Defense Information System Network (DISN); interfaces for Government, industry, and academia; satellite and other remote communications links; local, regional, and global control centers; and megacenters. The DII is an evolving infrastructure, for which the operational target date is the year 2000. A complete discussion of DII architecture, applications, and services can be found in DISA's *Defense Information Infrastructure (DII) Strategic Enterprise Architecture*.

[1] *Defense Information Infrastructure (DII) Strategic Enterprise Architecture*, DISA, Coordination Draft, May 31, 1995, pages 1-2.

A variety of other definitions of an open system, along with a discussion of standards and standards profiles, can be found in Section 1 of the *Next Generation Resources (NGCR) Acquisition Guide*.

2.2 GUIDING PRINCIPLES OF THE OPEN SYSTEMS ENVIRONMENT

"An Open System Environment encompasses the functionality needed to provide interoperability, portability, and scalability of computerized applications across networks of heterogeneous, multi-vendor hardware/software/communications platforms. The OSE forms an extensive framework that allows services, interfaces, protocols, and supporting data formats to be defined in terms of nonproprietary specifications that evolve through open (public) consensus-based forums."[2] Open systems with their set of applied standards are intended to function efficiently in the OSE. A well-developed and deployed OSE also supports data sharing and software reuse as well as cross-functional requirements.

The TAFIM provides the sound guidance and basis for evolving the OSE framework, which requires that the following OSE characteristics be incorporated in the engineering and design of C4I and information systems:

- **Standards-based** - importance of standardized data, interfaces, and architecture.
- **Portability** - capability to move from one environment to another through use of standardized data and interfaces, common languages, etc.
- **Scalability** - capability to move from one environment to a smaller or larger environment (including increased/decreased data flows) through use of standardized data and interfaces, common languages, etc.
- **Interoperability** - capability to communicate and operate with disparate systems within and outside of the primary operating environment through use of standardized data, interfaces, and architecture.

These characteristics are considered to be the basic "guiding principles" that program managers should take into consideration in planning and managing their programs. The program management areas where OSE principles should be of concern to the program manager are described in Section 3. The relationships of the OSE principles to the program management areas and guidance that may assist the program manager in assuring that these principles are properly addressed and incorporated in technical program activities are provided in Section 4.

[2] *Guide on Open System Environment (OSE) Procurement*, Gary E. Fisher, NIST Special Publication 500-220, October 1994, page iii.

This page intentionally left blank.

3.0 AREAS OF OSE CONCERN IN C4I AND INFORMATION SYSTEMS PROGRAM MANAGEMENT

Program management in the DoD can be defined as a systematic, coordinated process for selectively and collectively accomplishing the technical and managerial functions necessary to attain the timely, effective, and efficient acquisition and operation of systems and services. This section reviews the planning and implementation of program management process activities and products in which OSE principles and standards should be incorporated. The emphasis is on the program management of major system acquisitions; however, the same management principles and functions should apply to all C4I and information systems acquisitions, regardless of size. Modified management approaches and instructions unique to each Service may also apply, although the aspects of a program that must be demonstrated should be identical.

References to the DoD directives, standards, and other guidance documents, including the TAFIM, that contain complete direction and the recommended management approaches for subject area implementation are provided in each program area write-up. (Appendix C contains the complete listing of all references used.) These references should be reviewed if more in-depth information is required in a particular program management area. Also, Appendix F contains a listing of DoD services that can provide additional information or guidance in a particular subject area. A consolidated view of the program management activities discussed in this section, including the products to be produced and the management responsibility, is provided in Appendix G.

3.1 FUNCTIONAL PROCESS IMPROVEMENT

Functional process improvement (FPI) is an iterative management process by which information management in the DoD is defined and evolved. Although not formally considered a part of the life-cycle management (LCM) process, the FPI process precedes the initiation of the LCM process and eventually feeds most programs into the LCM process once system initiatives are identified and defined. FPI involves the streamlining and standardization of current processes, data, and C4I and information systems across the DoD. As depicted in Figure 3-1, FPI begins with the elimination of non-value-added activities and continues through rigorous analyses to identify changes in the way missions and functions are accomplished. It is through the FPI process that a mission need is defined or revised and C4I and information systems are developed or modified.

The Office of the Secretary of Defense Principal Staff Assistants (OSD PSA), along with the Chairman of the Joint Chiefs of Staff, has overall responsibility and authority to define DoD functional requirements and evaluate and improve current processes, data, and the supporting C4I and information systems. Direction, requirements, and guidelines for FPI are contained in DoD 8020.2-M (Draft) and 8020.2-M, Change 1, which establish the process improvement responsibilities and procedures for all DoD areas and activities. DoD 8020.1-M also provides

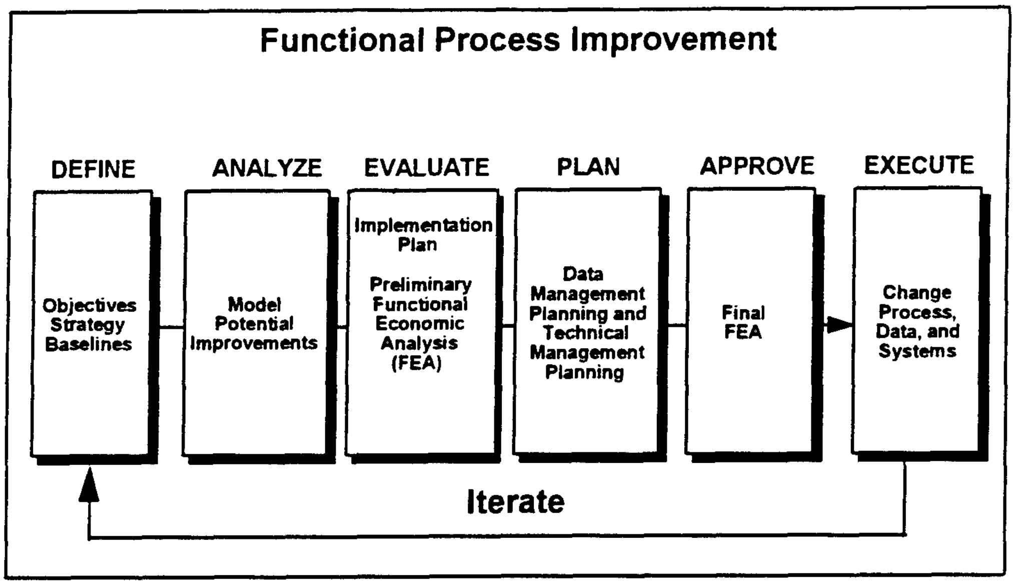

Functional Process Improvement
DEFINE
ANALYZE
EVALUATE
PLAN
APPROVE
EXECUTE
Objectives Strategy Baselines
Model Potential Improvements
Implementation Plan
Preliminary Functional Economic Analysis (FEA)
Data Management Planning and Technical Management Planning
Final FEA
Change Process, Data, and Systems
Iterate

Figure 3-1. Functional Process Improvement Process

information on the services and support mechanisms available to assist in performing FPI. The services provided by the Defense Information Systems Agency (DISA) are identified in Appendix F of this document. The *Acquisition and Technology (A&T) Architecture Development Handbook (Draft)* is an additional information source identifying the relationships and links between the FPI process and the standards-based architecture (SBA) process[1] - a process that intersects with and supports the development of the FPI-required products (e.g., Corporate Information Management Implementation Plan, Functional Area Strategic Plan, Baseline Analyses, Functional Economic Analyses, Functional Architecture) produced during the FPI process. A description of the SBA process can be found in TAFIM Volume 4.

3.2 MIGRATION PLANNING

Migration planning involves assessing the functional, technical, data, and programmatic dimensions of C4I and information systems within a functional area and determining the future of those systems identified as migration systems. In this respect, the purpose of migration planning is to identify systems that best meet functional area requirements and support improvement initiatives in processes, data, and infrastructure. This includes assessing and eliminating systems where duplication of functionality exists, assessing new technology and best practices, selecting standard systems (i.e., migration systems), conducting a detailed assessment of supporting infrastructures, developing acquisition and integration strategy, developing an implementation strategy, and developing and deploying the systems. Products of migration planning may include Integration Decision Papers and Technical Integration Plans, influenced

[1] The SBA Process guides the application of the technical architecture framework and provides a standard methodology for the development of technical architectures.

by Functional Economic Analyses (FEA) developed during the FPI process (see Section 3.1), and migration strategies and plans.

A more precise description of migration planning, including the requirements and responsibilities for this activity, are contained in DoD 8020.2-M (Draft) and DoD 8020.2-M, Change 1. TAFIM Volume 4, *DoD Standards-Based Architecture Planning Guide*, also provides a methodology for planning and implementing system migration as part of the SBA process. The SBA process depicted in the guide is an effective means of performing migration planning activities and can assist an organization in advancing selected migration systems toward the target architecture of all selected systems identified for the organization and feeding service requirements to the DII.

3.3 REQUIREMENTS

The requirements engineering phase of the life-cycle is recognized as one of the most important phases. Decisions made during this phase can have a significant impact on design, its implementation, integration, and testing. Program managers must be aware of the importance of this phase and the relationships among the different types of requirements and their impact on the program and system baselines. An understanding of these relationships, or the lack thereof, can have a significant impact on the cost and schedule of any program.

Depending on need and schedule, an acquisition or development manager can build a system in isolation (i.e., unfettered by policy or directives). More traditionally, the program manager considers the DoD policies, directives, acquisition guides, etc., when developing the system. A third scenario brings in all the former requirements and, in addition, takes into consideration adjunct requirements. The emergence of adjunct requirements (i.e., requirements that are levied on a program and are external to the system's set of performance requirements) can present added constraints or demand additional resources in the development process. Typically, adjunct requirements are not fully understood, defined, or considered in the conceptual or early life-cycle phases. Their impact will become evident in the development phase and more significant during implementation. Systems can be developed in the absence of adjunct requirements and still meet the intended set of operational and performance requirements; however, their inclusion in a development can represent significantly added scope.

An increasing demand for systems deployment in complex operational scenarios containing cross-functional interfaces and requiring conformance to Open System principles results in the creation of adjunct requirements. Introducing new technologies into a development can further increase the set of adjunct requirements. Adjunct requirements also require a framework for implementation and are needed to define a complete application portability profile. Program managers will be affected by adjunct requirements if their systems are required to implement in a particular DoD mandated language (e.g., Ada); utilize reusable components (e.g., design, architecture, software); adopt certain standards or methodologies (e.g., ICAM Definition Method [IDEF], object-oriented); utilize a particular environment or tool set (e.g., Computer-Assisted Software Engineering [CASE], Integrated Computer-Assisted Manufacturing [I-CASE]); procure from a standard set of defined resources (e.g., hardware, instruction set, chip set); adopt standardized components and/or security elements (e.g., operating system, compartmented mode workstation, database); and incorporate or introduce a new technology previously excluded. The degree of impact on a program will depend on the life-cycle phase in which the adjunct requirement is introduced and on the type of resources required to implement it. Adjunct requirements generated from these activities can result in added schedule or cost, unless their impact is understood and planned for early in the life-cycle.

Policies, directives, orders, and guidelines also directly drive or influence a manager's program. They establish a direction that must be conformed to and a set of schedule milestones that DoD management will monitor. They represent higher order constraints or mandates that affect the entire life-cycle. These key policies and directives are considered as pseudo-adjunct requirements, since they are recognized and understood by program managers and are planned for as an integral part of the acquisition and development process.

Figure 3-2 shows an optimum Requirements Model including adjunct requirements (i_1 and i_2 are iterations). A traditional Requirements Model is depicted in the three central boxes of Figure 3-2. The traditional model shows user requirements driving system requirements, which in turn drive the derived and allocated requirements. These requirements, in turn, are driven (or at least affected) by policy, directives, and orders, also depicted in the figure. As a system becomes more complex and as users become more sophisticated, the need for more constraining or modulating requirements will typically arise; the Requirements Model takes on a corresponding level of complexity from the introduction of the adjunct requirements. The introduction of adjunct requirements forces the model to become more of a process, in which the application of adjunct requirements necessitates further interaction between the requirements themselves and iterations of the process.

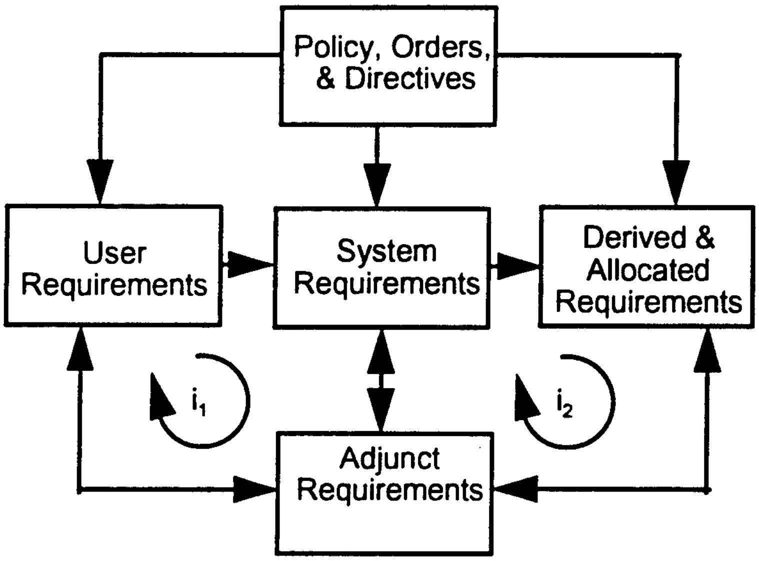

Policy, Orders, & Directives
User Requirements
System Requirements
Derived & Allocated Requirements
i1
i2
Adjunct Requirements

Figure 3-2. Requirements Model

The model is provided to make the program manager aware of the need to plan judiciously based on program needs and an extended set of requirements (i.e., the adjunct requirements). The model should assist in the development of a disciplined requirements process, which is necessary for the orderly translation of incomplete and informally identified user requirements into formalized, traceable system requirements.

A well-defined requirements process enables the development of appropriate requirements models to assist in this definition and refinement. Furthermore, such a requirements process will enable a separation or clear distinction between system prototypes (intended to optimize the design relative to requirements), and a requirements model (intended to define and mature system requirements). This distinction between models and prototypes will subsequently enable the synthesis of design derived directly from executable specifications in support of these prototypes and generated automatically by CASE tools or other design automation aids.

3.4 DETERMINING MISSION NEED

For C4I and information systems, mission need determination begins when the functional user identifies deficiencies or shortfalls in existing defense capabilities, identifies technological opportunity, or determines more cost-effective means of performing assigned tasks within the mission area. The functional user further defines or revises the perceived mission need through functional process review and information needs analyses, during which time alternatives to new development, use of commercial or existing systems, or tactics changes that may satisfy the existing or emerging need are considered and identified. When no other alternative is available, a Mission Need Statement (MNS) is developed to summarize the results of the analysis process and to document the mission need leading to the development of a new or modified C4I and information system. Approval of the MNS at Milestone 0 starts the life-cycle management process and establishes the program for system development or modification.

3.4.1 ·Mission Need Statement

The MNS defines and documents a mission need and justifies resource expenditures to identify and explore alternative solutions or system design concepts. At a minimum, the MNS describes the current organization and operational environment, with emphasis on existing functional processes, and identifies deficiencies in existing capabilities, new or changed functional requirements, and/or opportunities for improvement. It also addresses constraints and assumptions for functional, technical, and financial areas that may have an impact on potential alternative solutions; the relationships of the identified need to the current Corporate Information Management Strategic Plan[2] and Enterprise Integration (EI) Implementing Strategy[3] and functional area strategic planning and direction; the system location and general schedule for the

[2] *Corporate Information Management for the 21st Century, A DoD Strategic Plan*, ASD/ C3I, June 1994

[3] *DoD Enterprise Integration (EI) Implementation Strategy*, DISA Center for Integration and Interoperability, June 1994

implementation and deployment of the new or modified functionality; and any cooperative opportunities, such as a program addressing a similar need at another DoD or federal organization or within an allied nation.

The functional user prepares the MNS in accordance with DoD 8120.2-M, Part 2, and submits it for validation and approval in accordance with DoD 8120.2 paragraphs E.2.b, E.2.c, and E.8.e. The appropriate OSD Principal Staff Assistant and the Chairman of the Joint Chiefs of Staff, or a designated representative, validate the initial MNS, depending on the acquisition category of the program (i.e., major versus nonmajor system). The appropriate Milestone Decision Authority (MDA) approves the validated MNS at Milestone 0. The complete MNS may be updated, if appropriate, and revalidated for each milestone review subsequent to Milestone 0. It is also updated, if appropriate, and revalidated at the time a C4I and information system is designated as a migration system. DoD 8120.2 and DoD 8120.2-M provide further guidance on MNS validation and approval. Additional information regarding the milestone review process is provided in Section 3.12.1.

3.5 STANDARDS AND STANDARDS PROFILES

Standards are the complete, consistent suite of guideline documentation that reflects common consent among the organizational bodies on products, practices, or operations. Their primary purpose is to control the variability of products and processes. For example, information technology standards provide technical definition for processes, procedures, practices, methods, materials, items, engineering practices, operations, services, interfaces, connectivity, interoperability, information formats, content, interchange, transfer, and other standardization topics. They are also the basis for all life-cycle decisions affecting interoperability, portability, and scalability and are essential in achieving Open Systems design.

To ensure the intended compatibility, interpretability, and integration of C4I and information systems, IT standards planning and the documentation of selected standards are mandated by the DoD 8120 series of life cycle management directives and the TAFIM. This DoD policy clearly stipulates that all C4I and information systems programs are required to accomplish standards planning, including the identification of information technology profiles, in accordance with the TRM for Information Management, previously discussed in Section 2 and fully described in TAFIM Volume 2. In this respect, each program is required to prepare and produce an IT standards profile beginning no later than Milestone I, with future updates, thereafter, in each system life cycle phase. The standards profile is required for inclusion in the System Decision Paper (SDP) submitted, by the program manager, for each milestone decision. It also accompanies the Test and Evaluation Master Plan (TEMP) at Milestones II, III, and IV for standards conformance test planning purposes.

3.5.1 Applying the TRM to Standards Profiles

A knowledge and understanding of the TRM, discussed in TAFIM Volume 2, provides the insight needed to develop and identify standards/standards profiles, support environments, migration strategies, and technology issue resolution, since the TRM is a mechanism for establishing relationships/linkages between service areas, the services themselves, and standards. Establishing these linkages provides the basis for selecting environments and their services to ensure interoperability. It also provides the basis for prioritizing tasks/acquisition components and standards as a function of the life cycle and "best time to effect." The latter is equivalent to the emerging concept of "just-in-time engineering/manufacturing" used to reduce inventories and maintenance costs.

Knowledge of the TRM, service areas and services, and the available standards identified in the AITS and ITSG mentioned above also contributes to the effective planning and implementation of acquisition strategies and program activities. By establishing relationships and mappings of standards to services and service reference models (e.g., NIST/ECMA Special Publication 500-211), a program manager can select tools in an ordered and prioritized manner, precluding a costly initial investment in those tools, that can be obviated by technology transfer rates offering increased functionality and capability in next-generation products and environments.

3.5.2 Developing Standards Profiles

A standards profile is a defined set of one or more standards, and where applicable, the identification of chosen classes, subsets, options, and parameters of those base standards necessary for accomplishing a particular function. The standards profile may contain a set of one or more base standards, along with specific subsets, classes, options, and parameters necessary to accomplish a particular function. The specific profile becomes part of the program documentation baseline and matures with the system design as the program progresses through each life-cycle phase. The requirements specified within the profile are included in systems acquisition documentation as performance requirements, functionally allocated to, and integrated appropriately into program and contract documents, such as specifications, Statements of Work (SOWs), proposal evaluation criteria, proposal instructions and formats, and contract data requirements.

TAFIM Volume 7, *Adopted Information Technology Standards (AITS)*, provides architects and system planners with the definitive set of IT standards for standards profile development. Implementing activities are encouraged to select from this repertoire of standards to meet the needs of specific mission areas. Use of these standards will help provide a consistency across the enterprise, mission, function, and applications levels of the DoD Integration Model, as described in TAFIM Volume 1, and will enable program managers to guide their programs toward a collective DoD OSE.

A companion document to TAFIM Volume 7 to be used in the selection of standards and the development of standards profiles is the *Information Technology Standards Guidance* (ITSG). The ITSG is the foundation document for the AITS. It provides amplifying implementation guidance for those standards identified in TAFIM Volume 7 as well as supporting information on AITS standards hierarchies. The ITSG also includes information on related or emerging standards precluded from the AITS, and recommendations for specifying standards in system acquisition documentation. Because of the ever-constant changes in standards, the program manager should also monitor Government and industry trends and keep abreast of ISO, IEEE, ANSI, etc., and new developments in preparing standards profiles.

The Center for Standards, within DISA and responsible for the evolution of IT standards policy, will provide customer assistance in applying the information found in the AITS and ITSG. Users of AITS and ITSG information are encouraged to contact the Center for Standards for assistance or to identify functional requirements and/or standards not yet incorporated in these documents. (See listing for Center for Standards in Appendix F.)

3.6 DATA ADMINISTRATION, DATA MODELING, AND DATA STANDARDIZATION

Data administration is the function that oversees the management of data across all facets of an organization and is responsible for central information, planning, and control. Department of Defense Directive (DoDD) 8320.1, *DoD Data Administration*, establishes the policies for the administration of data in the DoD and authorizes a DoD Information Resource Dictionary System (IRDS) as a primary tool of data administration. As discussed in DoDD 8320.1 (Enclosure 3), the responsibilities of planning, managing, and regulating data are assigned to the DoD Data Administrator (DoD DAd), located within the DISA Center for Software (see Appendix F). The DoD DAd implements and manages DoD-level data administration policies and procedures and supports the development and management of useful, available, and accessible information to enable the successful execution of the mission of the Department. The DoD DAd also tracks all the entities and data elements that represent the emerging DoD standard information requirements and provides the technical infrastructure for data administration, including the DoD Data Model, the Defense Data Dictionary System (DDDS), and procedures for data modeling, data standardization, data security, data quality assurance, and database operations.

The DoD DAd has enacted the Defense Information Management Program, which requires that accurate and consistent information be available to decision makers for the effective execution of DoD missions. The program operates with the following objectives in mind:

- To develop the DoD Enterprise Data Model (EDM) to depict overall DoD mission needs and support operational capabilities requiring the collection, storage, and exchange of data.
- To develop data elements for standardization through data modeling efforts.
- To create a base of shared information through the DoD EDM and standard data structures and elements. This will enable functional and technical personnel to perform their tasks in an integrated, effective, and efficient manner.

- To implement data administration aggressively in ways that provide clear, concise, consistent, unambiguous, and easily accessible data DoD-wide.
- To standardize and register data elements that meet the requirements for data sharing and interoperability among C4I and information systems throughout the DoD.
- To use applicable federal, national, and international standards before creating DoD standards or using common commercial practices.

Each DoD Functional Area assigns a Functional Area Data Administrator (FDAd) to implement data administration procedures and serve as the functional area representative on functional issues affecting DoD data administration. The FDAd also identifies data administration resources needed in the Functional Area and identifies functional requirements for submission to the DoD data administrators.

Component Data Administrators (CDAd) are assigned to help implement data administration procedures across all functional areas within the Component. They identify the interface between the users, database administrators, and application developers of the C4I and information systems within the DoD Component and ensure Component adherence to DoD data administration policies, procedures, and standards.

The uniform management and operating procedures established for use by all DoD levels in managing and implementing DoD data administration activities and products are found in DoD 8320.1-M, *Data Administration Procedures*. This manual implements the data administration program established by DoDD 8320.1 and provides the mission, goals, benefits, and concept of operations of the data administration program; the roles, relationships, and responsibilities of the DoD data administration community; program management procedures for sustaining the data administration function; and procedures for maintaining and using a technical infrastructure.

3.6.1 Data Modeling and Standardization

A data model is the graphical and textual representation of data a business needs to accomplish its mission. It is a representation of data objects that can be shared and reused across application systems, organizational boundaries, and different functional areas. Models provide information about the interests of an enterprise; facilitate improvements in strategies, tactics, and operations; provide a basis for database design; facilitate an understanding of data leading to the identification of sharing possibilities; and reduce redundant data entry and unintentional replication of data. The basic steps of DoD data model development include data model reviews by data administrators at all DoD levels to ensure data standardization, which promotes data sharing, software reuse, and, most importantly, interoperability. These reviews ensure the proposed entities, attributes, and relationships identified in the data model adhere to mandatory technical and functional requirements and are representative of the DoD-wide data standardization perspective provided in the DoD EDM.

The DoD EDM is the integrated view of the data requirements of the functional areas and Components in the DoD. It is developed and continuously extended based on reviews of data models developed to document data requirements across DoD functional areas. It is also the infrastructure to support the DoD data administration objectives. DoD C4I and information systems that are to conform to DoD data administration procedures are to be developed in this DoD-wide perspective, through the use of modeling tools and standard metadata. The manual, *DoD Enterprise Data Model Development, Approval, and Maintenance Procedures* (DoD 8320.1-M-x), is interim guidance for developing data standards that are to become part of the EDM. This manual should be used in conjunction with DoD 8320.1-M-1, *Data Element Standardization Procedures*, in the development, approval, and maintenance of EDM-related products.

DoD 8020.1-M (with Change 1), *Interim Management Guidance on Functional Process Improvement*, provides additional guidance on data modeling, while TAFIM Volume 4 (and its associated *A&T Architecture Development Handbook [Draft]*) provides methods for identifying opportunities for data improvement, when exploring business improvement opportunities. A process for developing data requirements and shared information approaches can also be found in Section 4 of the working draft of the *Acquisition and Technology (A&T) Corporate Information Management/Enterprise Integration (CIM/EI) Program Management Structure*.[4] A wide array of information on data modeling and standardization is also available from the DISA Center for Software (see listing of services in Appendix F), responsible for the promulgation of the aforementioned policy on data standardization and modeling and the maintenance of the EDM. The Center for Software also operates and maintains the DDDS discussed in the following subsection.

3.6.2 Defense Data Dictionary System

The Defense Data Dictionary System (DDDS) is a centrally controlled, DoD data repository put in place and managed by the DoD DAd to receive, store, support access to, and manage standard data definitions, data formats, usage, and structures (e.g., architectures, subject area models, and other data model products). Specifically, the DDDS is to assist the DoD in creating and maintaining a repository system in the following ways:

- Collect and store standard elements and their attributes
- Identify DoD organizations and processes using standard elements as defined in information models
- Provide convenient, on-line data element documentation query and reporting capabilities throughout the DoD
- Provide the capability to track the state of each standard element throughout its life-cycle, from its proposed candidacy through its archival and deletion
- Provide the capability to identify the impact of proposed changes on standard elements.

[4] Provides a framework and uniform management structure for implementing the CIM/EI program within the A&T community.

The DISA Center for Software should be contacted for further information and guidance on DDDS services (see Appendix F).

3.7 ESTABLISHING ARCHITECTURES FOR OPEN SYSTEMS

An Open Systems architecture depicts a system in which the components, both hardware and software, are specified in an open manner. In establishing an open system architecture, the Program Management Office (PMO) must determine the needs and functional requirements to be fulfilled by the system through the in-depth analysis of:

- **Target system requirements** - including data, communications, hardware, security, applications, etc.

- **Existing infrastructure**- including wide area networks (WANs), local area networks (LANs), servers, routers, communications, applications, etc.

These analyses are then used to identify integration needs and evaluate integration issues. The program manager must be cognizant of all developments above the program level (i.e., enterprise, mission, or functional area level) in regard to the open architecture, as it is a "living" and "dynamic" entity. The functional requirements must also be applied across the various open hardware and software standards to meet the system requirements. The use of open standards allow product choices with compatible interfaces that can be combined to create an open system architecture. The use of standards and common functional and technical architectures contributes to standard, portable, scalable, and interoperable systems for which individual components can be acquired and configured, by different executive agents, over an extended period of time. Within the umbrella of common architectures, data, applications, and infrastructures can be managed according to their separate life-cycles and integrated into complete systems.

There are a variety of architecture models to choose from in the establishment of functional and technical architectures for C4I and information systems. Each has its advantages and disadvantages, and each must be evaluated in light of the system requirements and environment (i.e., open, legacy, or migration). Components may be mixed and matched from the various architecture models, as long as services are allocated per the Technical Reference Model and as long as a standards profile is adhered to. Architecture concepts and design guidance for use in establishing an architecture are contained in Section 3 of TAFIM Volume 3. The preferred methodology for planning and implementing an architecture is presented in TAFIM Volume 4, *DoD Standards-Based Architecture Planning Guide*. DISA's *Architecture Relationships and Definitions* should be used in order to become familiar with the basic architecture concepts. Also, a close association with DISA should help ensure that the program is on track with recent developments.

3.8 SYSTEM SECURITY

In each C4I and information systems endeavor, program management and staff must consider security at all levels and throughout the system life-cycle to provide multifaceted, cost-effective protection of the data being processed or transmitted. A security program with basic principles and safeguards that assure data confidentiality, reliability, accuracy, and availability, and that maintains accountability for actions within the operational environment should be fundamental to the design, implementation, operation, and maintenance of the system. This concept allows for confidentiality that limits data access to individuals with a need to know; reliability that data are not altered and results are accurate; availability that assures data are on hand when needed; and accountability that audits activities for responsibility of accomplishment.

The inclusion of information systems security throughout the planning and development process provides for cost-effective fielding of systems that are legal and regulatory-compliant. Accordingly, legal and regulatory guidelines have evolved to govern Federal Agency and Department information security operations. These guidelines range from Public Law 100-235, the Computer Security Act of 1987 and its implementation instruction (Office of Management and Budget [OMB] Circular 90-08), to National Computer Security Center (NCSC) directions, the "rainbow series", and Departmental regulations (i.e., DoDD 5200.28, DoD 5200.28-M, DoD-Standard (STD)-5200.28-STD, DoDD 5200.5, DoD 5200.1-R, and DoD 8120.2-M), which require the preparation of a System Security Policy and System Security Plan for milestone decision review.

Conformance to Open System requirements also adds a layer of complexity to security concerns. In an Open System, secure data are potentially accessible to more users than in a closed system. Special attention should be paid to emerging protocols, multilevel security schema, etc. Although the specification and application of security standards does not totally ensure a secure system or design, the program manager must be sure that security engineering is performed with the most current standards in mind and in accordance with the DoD Goal Security Architecture (DGSA), a primary consideration in establishing a security structure for C4I and information systems. The DGSA is an evolving, generic security architecture, developed by the DISA Center for Information System Security (CISS), under the Defense Information Systems Security Program (DISSP), a joint undertaking of DISA and the National Security Agency (NSA). TAFIM Volume 6 addresses the security requirements of the DGSA and the process by which organizations can identify the specific security requirements of their missions. In brief, the DGSA specifies the security principles, concepts, functions, and services that target security capabilities to guide system architects in developing their specific architectures. It also includes a generic security architecture that provides an initial allocation of security services and functions. Program managers should become familiar with the DGSA, as described in TAFIM Volume 6, and with the other applicable security guidance mentioned above, to assure legal and regulatory compliance with DoD and federal security guidelines and initiatives.

The Center for Systems Engineering within DISA is responsible for the development of TAFIM Volume 6 and can be of assistance in providing additional information and guidance on the DGSA. The Center for Systems Engineering is listed as a resource in Appendix F.

3.9 ESTABLISHING THE PROGRAM MANAGEMENT TEAM

The key to a successful program is to establish a management structure that reflects the mission of the organization yet remains flexible enough to accommodate the needs of the program. The organization and management of the program should also be consistent with the importance and scope of the program. To comply with the C4I and information systems LCM policy and guidance in the DoD 8120 series of directives, a C4I and information systems program manager must be assigned at the beginning of the LCM Phase 0, Concept Exploration and Definition, in time to explore alternative system design concepts. The program manager is selected based on the level of education, training, experience, and other qualifications required of program managers, as specified in DoD 5000.52.M, *Career Development Program for DoD Personnel Manual*. The program manager ideally is a multidisciplined, experienced manager with sufficient tenure and interest in the program to provide continuity and establish accountability for program actions. The individual should be capable of establishing a program structure and program work force that compliments project size and technical complexity and should be knowledgeable about and capable of managing the programmatic and technical elements identified in the program structure.

The program manager should also be aware of the current topics of emphasis found in congressional testimony, DoD policy statements and speeches, and in the media, since some of these topics attain permanence by being incorporated into DoD directives or instructions. Most important, in managing the design and development of an Open System, the program manager must understand the functional and technical architecture framework in which the assigned system will perform and must be willing to enforce standard practices in all management and technical processes.

3.9.1 Program Management Charter

Program objectives are developed that set forth the capability in terms of mission need, cost, and schedule goals being sought by DoD upper-level managers when establishing the requirement for new or modified C4I and information systems. These objectives are communicated to the program manager by the DoD management authority (i.e., Deputy Secretary of Defense, or designated authority, etc.) in a written charter that serves as a contract between the program manager and the chartering authority. In addition to program objectives, the program manager's charter defines the authority, organization, resources, responsibility, scope, and methods of operation of the C4I and information systems program, as well as the lines of authority and accountability. The charter is prepared and processed in accordance with the policy, instructions, and procedures contained, respectively, in DoDD 8120.1, Department of Defense Instruction (DoDI) 8120.2, and DoD 8120.2-M.

3.9.2 Program Management Team

A responsibility of the program manager is to recruit a staff or identify a program management team with the requisite skills and experience to manage the assigned system. In putting together a team for an Open Systems project, the personnel requirements for the team should be determined based on the work identified in the contract, specifically in the SOW and in the Contract Data Requirements List (CDRL) discussed in Section 3.14. The Work Breakdown Structure (WBS), discussed in Section 3.13 and linked directly to the SOW, is also a source for determining team skill requirements, since it defines the work to be accomplished and assigns resources and responsibilities to the work elements identified. Resource requirements may also be determined from the results of market and trade studies discussed in Section 3.11.

The most critical work elements in accomplishing OSE objectives are the technical engineering management organizations established within a program. These organizations, individually or as a whole, are the program manager's front line with the user. The effectiveness of these organizations depends on how well they are institutionalized in the program and how cognizant and sensitive they are to Open Systems issues and TRM service areas and views pertaining to architecture and standards. The leadership and control implications of these program elements are driven by the program size, program maturity (life-cycle phase), number of system segments, interface complexity, and individual skills. A generic technical engineering management structure for a development and integration type effort, however, is typically organized under the guise of systems engineering management. This organization may include all or some of the following types of personnel, with all or a mixture of the skills described:

- **Systems manager (chief engineer).** Lead technical manager who controls the architecture and all project-level engineering plans. Also manages the project's technical baseline and speaks for the program manager on technical issues. Has leadership skills, communication skills, a generalist perspective; pays attention to detail; and has a broad project experience in the areas of engineering, development, and test. Should report directly to the program manager.

- **Systems architect.** Plays a subordinate role to the systems manager and is responsible for the "vision" of the system, as stated in user requirements and desired expectations. Guides the development process from "cradle to grave." Is a participant in requirements development; is responsible for high-level systems design; and guides the design and test process. Has a sense of vision, communication skills, and the ability to work at the abstract level.

- **Systems engineer.** Plans, manages, and monitors all systems engineering activities. Develops and maintains systems functional, developmental, and operational "test-to" requirements. Analyzes requirements and allocates to system design. Identifies and allocates derived requirements within specialty engineering domains. Has leadership skills and broad engineering experience, with an ability to pay attention to detail. Should report directly to the systems manager or systems architect.

- **Systems test manager.** Plans/monitors all verification activities and is responsible for system integration and requirements compliance verification, including configuration item acceptance testing, item-to-item integration and checkout, system-level test (including external interface test), and system regression testing. Has systems engineering experience, communication skills, development experience; and pays attention to detail. Should report directly to the program manager.

- **Quality assurance manager.** Is the program manager's independent review authority. Ensures that project processes are being followed, including the management of project metrics, and audits for requirements compliance. Has standards and policy awareness, considerable systems engineering skills and experience; is process-centered with continuous improvement awareness; and has a broad project perspective. Should report directly to the program manager.

- **Configuration management (CM) manager.** Determines and coordinates all CM activities, including configuration control board activities; determines and monitors contractual CM requirements; establishes relationships with interfacing CM organizations; and ensures continuity and that uniform CM practices and procedures are followed. Like the quality assurance manager, is aware of standards and policy; has considerable systems engineering skills and experience; is process-centered with continuous improvement awareness; and has a broad project perspective. Should report directly to the program manager.

- **Systems engineering personnel.** Perform/monitor requirements analysis, system design, and system test planning functions during the initial phases of the project. Possible transition to verification and operational support tasks (testing, tech manuals, installation, and checkout, etc.) following approval of the critical design. Should report to the systems manager or systems architect.

- **Engineering specialty engineers.** Specialty engineering includes domains that require detailed expertise beyond the scope of the typical engineer or developer and including those engineering disciplines that influence system design, development, and operational support of a product, such as reliability and maintainability engineering, performance engineering, risk management, human factors engineering, safety engineering, life-cycle cost analysis, and logistics engineering Specialty engineers with specific expertise are typically integrated into a program to

  - Analyze and recommend engineering specialty requirements
  - Tailor standards and specifications to meet specialty requirements
  - Develop contract SOW input, specification input, and deliverable requirements
  - Evaluate offerers' responses
  - Prepare detailed specialty engineering management plans
  - Review development contractors' deliverables
  - Evaluate contractors' progress/conformance at design reviews
  - Monitor tests and conduct specialty tests
  - Evaluate operational performance
  - Evaluate engineering change proposals (ECPs).

Each engineering specialty should be part of the systems engineering organization during the initial phases of a program but may spin off or migrate from the systems engineering domain to become its own entity as development progresses.

3.10 DETERMINING PROGRAM STRATEGY

The program strategy is a combination of business and technical management concepts designed to achieve program objectives within imposed resource constraints. It is the method utilized to project design, development, and deployment requirements for the C4I and information systems and is the basis for formulating the acquisition plan and subsequent functional program plans, which guide the C4I and information systems program throughout its life-cycle.

The program manager formulates the program strategy during the concept exploration and definition phase of the LCM process and incorporates it in the Program Management Plan (PMP) for approval at the Milestone I review. DoDI 8120.2 identifies and describes four program strategies that may be considered: grand design, incremental, evolutionary, and other. The PMP preparation guidelines provided in DoD 8120.2-M identify the specific requirements for documenting the chosen strategy.

Government and contractor objectives should be clearly stated in the program strategy, as should the level of competition, estimate of contract value, type of contract, time phasing, and program incentives. It is also the program manager's responsibility, by means of the program strategy, to remain consistent with basic LCM policy but to tailor the LCM phases, activities, and milestones (see Section 3.12) to best fit the unique requirements and conditions of the program. In this regard and depending on the selected strategy, the program strategy may recommend combined or repeated milestone decision points, as well as associated activities within a life-cycle phase, if required. The number of replicated decision points, as well as the manner in which the increments between decision points will be reviewed, is included in the initial program strategy at Milestone I. The program strategy may be updated or refined in the subsequent life-cycle phases; however, any modification must be approved by the MDA.

Program strategy should be refined by requirements for interoperability, scalability, and especially, portability. Some other considerations in formulating the program strategy may include the general OMB policy to rely on the private sector for proposing solutions to functional requirements and to use contracting as a tool in the acquisition process (see OBM Circular A-109), and other necessary considerations, which include the favorable and unfavorable lessons learned from similar programs; recognition of and accommodations for risks and uncertainties; the proper relationship of risk sharing between the Government and the contractor; the Government tailoring of specifications and standards in consonance with contractor efforts (the objective being to avoid nonessential constraints on contractors); the optimal use of Government laboratories in furnishing technical direction during system development; the use of Non-Developmental Items (NDI)/Commercial-off-the-Shelf (COTS) products in lieu of development; and the possible reuse of existing resources. Section 1 of the *Next Generation Computer Resources (NGCR) Acquisition Guide* provides a detailed discussion of the advantages and disadvantages of a program strategy that includes NDI acquisition.

3.11 EXPLORING ALTERNATIVES THROUGH MARKET ANALYSIS

Selecting the right products for an Open System Environment requires conducting a market analysis based on market surveys, technical risk analysis, supportability risk analysis, mitigation techniques, and life-cycle cost impact assessments. Information derived from market analysis becomes an economic driver for possibly reviewing (possibly revising) requirements, as well as planning, budgeting, and implementing system upgrades and support. The remainder of this section addresses market surveys, trade studies, and trade-off analyses, which are decision-making tools that can be used in determining and evaluating the current technology market and OSE product options.

Market surveys provide the rationale for make or buy decisions and provide information on technologies, existing products, market share commercial production practices, and industrial capabilities. The results of market surveys are incorporated into the requirements decomposition process and used in technology assessments.

Two types of market surveys are typically performed: the initial market survey and the market investigation. During the initial market survey, defined system requirements should be compared with features of OSE-compliant products. The objective of this survey is to establish an awareness of the marketplace and to determine what products are available as NDI. One of the most important first steps in conducting the initial survey is early communication of the requirements to the vendors identified (OEMs, their representatives, and their suppliers). Such information includes operating parameters for hardware and software, environmental constraints, interface and integration requirements, etc., that will allow each vendor to better answer questions about possible solutions to the requirements. The subsequent market investigation is conducted following the identification of potential product sources, as obtained in the initial market survey, to obtain more specific information on the product and source so that a final decision can be made.

Other types of evaluation open to a program manager in making program decisions are trade studies and trade-off analyses. Trade studies are performed typically by the contractor throughout development as an essential part of the systems engineering process. Trade studies are controlled by systems engineering to integrate and balance all design-for and engineering specialty requirements and to compare candidate hardware and software standards and products available to meet program needs. As a formal decision analysis method, trade studies are used to solve any complex problem that has more than one selection criterion and to provide documented decision rationale for review by a higher authority. These analyses are necessary for establishing system configurations and for accomplishing detailed design of individual components. The trade study method is equally applicable to budgeting, source selection, test planning, logistics development, production control, and design synthesis. Trade-off analysis also provides a structured analytical framework for evaluating a set of alternative concepts or designs. Trade-off analysis is typically used in source selection, but it can also be used when criteria for study or parameters are conducive to objective evaluation or amenable to a numerical performance measurement scheme.

Additional information on market analysis, specifically information on how to conduct market research and surveys, can be found in Section 6 of the DISA *Acquisition How To Guide*.

3.12 LIFE-CYCLE MANAGEMENT PROCESS

The system life-cycle consists of the interval from system inception through system disposal. All activity in the system life-cycle centers on the state of definition of the system configuration at any time in its life-cycle. The Department of Defense uses a systematic technical management process to control the system life-cycle, as promulgated in accordance with the DoDD 8120.1, *Life-Cycle Management (LCM) of Automated Information Systems*, DoDI 8120.2 *Automated Information System Life-Cycle Management Process, Review, and Milestone Approval Procedures*, and DoD 8120.2-M, *Automated Information System Life-Cycle Management Manual*. As depicted in the directives, the process includes five life-cycle phases (Concept Studies Decision; Concept Exploration and Definition; Demonstration and Validation; Development; Production and Deployment; and Operations and Support), with sets of phased activities and periodic reviews, including milestone decision reviews at Milestone 0, I, II, III, and IV. Each milestone review is conducted by the appropriate MDA, discussed in Section 3.12.1, to determine how well program requirements are being met and risks are being managed. The DoD Component acquisition executives, program executive officers (PEO), and program managers are charged with the responsibility of the programs under their control to provide the focus and management to develop, field, and support the programs to meet user needs. These managers must work closely with their various counterparts in the Office of the Secretary of Defense and the appropriate committees to ensure the program is ready to proceed from one life-cycle phase to the next.

The required program management activities to be accomplished in each LCM phase, including the essential program documentation required for milestone decision, are identified in the DoD 8120 series of directives mentioned earlier. The program documentation listed in DoD 8120.2-M, which provides the core procedures and content requirements for milestone decision documentation, are the primary means for conveying to the MDA a complete description of the program activities and program issues. The documentation is intended to reflect the accomplishment and/or current status of specific planning and analysis tasks to be conducted before each milestone review, and is a synthesis of the existing program plans and essential information prepared by the various program organizations to support and guide the system acquisition. Also, the systems engineering documentation identified in Section 6 of DoDI 5000.2 may be developed and submitted as appendices to the PMP, should program activities and complexity warrant the development of such documentation. The PMP and other program documentation required by DoD 8120.2-M, as well as the planning documents that may be required from DoDI 5000.2, Section 6, are depicted in the Program Management Responsibilities Matrix contained in Appendix G.

3.12.1 Milestone Decision Authorities and Reviews

Periodic, formal program reviews (either scheduled milestone decision reviews or in-process reviews) are required before a C4I and information systems program can advance from one LCM phase to the next. The purpose of each review is to give management a current status of the program and to allow management to provide additional guidance and/or give milestone approval for advancement to the next life-cycle phase.

The MDA is responsible for conducting the milestone review and is assigned based on the acquisition category of the C4I and information systems program (major verses nonmajor) as described in DoDD 8120.1. For major C4I and information systems programs falling outside the purview of the Under Secretary of Defense for Acquisition (USD[A]), the MDA is ASD (C3I), who is the DoD senior IM Official designated in accordance with DoD Directive 5137.1. This authority may be re-delegated to the lead acquisition authority, DoD Component head, DoD Component acquisition executive, or the

Senior IM official within the DoD Component. For nonmajor C4I and information systems programs, the DoD Component head is the designated MDA. This authority may also be further delegated to the appropriate lowest level, commensurate with the resources and risk involved.

The MDA performs formal program reviews in accordance with the LCM policy, responsibilities, process, and procedures of DoD 8120.1 and DoD 8120.2, and the uniform procedures for conducting LCM activities and preparing LCM documentation in DoD 8120.2-M. For non-major C4I and information systems programs, the MDA adheres to the various LCM policies and procedures established by the respective DoD Component heads and the OSD PSAs. Through the review and analysis of the LCM documentation required for MDA review, the designated MDA provides the C4I and information systems program manager and staff with the appropriate program direction. Milestone approval, conditional milestone approval, or approval of specified activities must be obtained before program management may proceed with activities in the next life-cycle phase. A review is successfully completed when the MDA makes management judgments on what program activities may be permitted and specifically authorizes those activities for next life-cycle phase implementation

3.12.1.1 The Defense Acquisition Board

The Defense Acquisition Board (DAB) is the oversight management mechanism for major Defense acquisition programs. It is the primary forum used by the DoD Components to resolve issues, provide and obtain guidance, and make recommendations to the Under Secretary of Defense for Acquisition on matters pertaining to the DoD acquisition system. Formal DAB reviews are conducted at each milestone to assess Service accomplishment of the previous phase and to assess readiness to proceed to the next phase of the LCM process. The USD(A) may also hold special in-process reviews between milestones, when warranted.

The USD(A), as the Defense Acquisition Executive (DAE), chairs all program and milestone decision reviews for major defense acquisition programs (DoDD 5000.1/DoDI 5000.2). To help the DAE conduct milestone reviews, four DAB committees (Strategic Systems, Conventional Systems, C3I Programs, and Major Automated Information Systems) have been established. These committees conduct pre-DAB reviews and develop, investigate, and resolve program issues.

3.12.1.2 The DoD Major Automated Information System Review Council

The DoD Major Automated Information System Review Council (MAISRC) is the life-cycle management review body for all major C4I and information systems subject to review under the policies and procedures of the DoD 8000 series Directives. It is composed of a chairperson, members, an Executive Secretary, and staff. ASD (C3I) chairs and operates the MAISRC (independently of the DAB) in resolving program issues and facilitating milestone decisions in the role of MDA. The MAISRC conducts milestone reviews to evaluate the completion of the minimum required LCM accomplishments and exit criteria; provides advice on program readiness to the MDA and recommends appropriate movement to the next LCM phase; determines the adequacy of proposed plans for subsequent LCM phases; and recommends exit criteria for each milestone review. (DoDI 8120.2 and DoD 8120.2-M should be reviewed for further details on this process, including the documentation required and specific responsibilities of the program manager and other review participants. Appendix G, however, does identify the overall MAISRC documentation required for each milestone review in accordance with DoD 8120.2-M.)

3.12.1.3 The In-Process Review

The MDA may call an in-process review (IPR) at any time within the life-cycle of a program to determine current program status, progress since last milestone review, program risk and risk-reduction measures, and potential program problems that require guidance. An IPR will also be called when there is a breach in the program baseline. As requested by the MDA, the program manager will be required to submit documentation for MDA review. The documentation is assembled from existing program management documentation and may be supplemented with additional documentation required to support specific issues to be addressed at the IPR.

3.12.2 The System Decision Paper

The System Decision Paper (SDP) is the principle document for recording the essential C4I and information systems information critical to the DoD decision-making process, such as mission need, alternatives, management approach, schedule, resources, issues, risks, security issues, and supporting rational and decisions. The SDP represents the functional and C4I and information systems program management coordinated position for the C4I and information systems and is the primary document supporting MAISRC process. The program manager prepares the initial SDP after Milestone I, with updated SDPs submitted thereafter for each subsequent milestone review. The SDP must be approved by the appropriate level at the completion of each LCM phase in order for the respective milestone to be achieved. Part 4, Attachment 1, of DoD 8120.2-M provides the procedures and the recommended format for preparing an SDP.

3.12.3 The System Decision Memorandum

The System Decision Memorandum (SDM) documents the milestone approval decision of the MDA, the guidance provided, and the exit criteria established for the next LCM phase, including the activities to be accomplished. The MDA prepares and signs the SDM following each milestone decision review.

3.13 PROGRAM PLANNING AND CONTROL

Planning establishes the framework upon which the program manager authorizes and issues work to the task organizations. Planning is evolutionary and continues through the life of the program. The planning process breaks the WBS requirements down into subordinate elements of work appropriate to the size of the program, schedules its accomplishment, establishes budgets, and allocates resources. The work authorization process is the means by which the program manager controls the flow of work, authorizes task organizations to perform the work, and establishes performance, budget, and schedule parameters. Planning the work also requires the definition of the technical effort and the requirements for labor, material, tooling, equipment, facilities, and funding.

In addition to the WBS, the acquisition strategy, PMP, and the requirements of the Request for Proposal (RFP), SOW, specifications, and other contractual documents provide the initial impetus for planning and organizing the total program. The work effort and requirements derived from these documents culminate in the development of the WBS and other management and planning documents such as the Work Package, the Program Master Schedule, associated authorization documents, and internal Government and contractually required functional plans, such as the Systems Engineering Management Plan (SEMP), Integrated Logistics Support Plan (ILSP), TEMP, SDP, Configuration Management Plan, etc., which lay out the details for the establishment and implementation of specific segments of the overall program effort.

The remainder of this section discusses the WBS and Program Master Schedule, two of the most important tools of the program manager, and the cost/schedule and control methods used in measuring program performance.

3.13.1 The Work Breakdown Structure

A Work Breakdown Structure (WBS) is a product-oriented family tree, composed of hardware, software, services, and data that completely defines a program. The WBS displays and defines the product(s) to be developed and/or produced and relates the elements of work to be accomplished to the end product. The WBS is the foundation for:

- Program and technical planning
- Cost estimating
- Schedule definition
- Statements of work and specification of contract line items
- Progress status reporting and problem analysis.

The WBS is essential in providing the capability for the program management office to exercise technical, schedule, and financial control of the program. It also serves as the framework for the contractor's overall management system.

Four basic types of WBS formats are identified in Military (MIL)-STD-881, the standard for the WBS, although other specialized WBS that suit particular applications during design and development may be used. The four basic WBS types prescribed by MIL-STD-881 are:

- Summary WBS
- Project summary WBS
- Contract WBS
- Project WBS.

3.13.1.1 Summary WBS

A summary WBS is a structure in which the upper three levels of the WBS are specified by MIL-STD-881. The structure has a uniform element terminology, definition, and placement in the family-tree order. Appendices A through G of MIL-STD-881 provide a three-level WBS for each of the seven types of material items procured by the DoD (i.e., aircraft systems, electronic systems, missile systems, ordinance systems, ship systems, space systems, and surface vehicle systems).

3.13.1.2 Project Summary WBS

A project summary WBS is derived from MIL-STD-881 but is tailored to the specific program. This WBS is also specified to three levels of detail. The project/program office builds the project summary WBS by selecting applicable elements from the example project summary WBS in MIL-STD-881. This is usually done at the beginning of concept exploration and definition phase (Phase 0) and is included in the RFP and finalized at contract award. From this WBS, the contractor can develop individual contract WBSs (see paragraph 3.13.1.3) in compliance with the instructions contained in the RFP. (A preliminary WBS is normally part of the contractor's proposal.) The RFP contract line items (CLINs), configuration items (CIs), SOW tasks, and contract specifications, are elements of the preliminary contractor WBS. A final contractor WBS will be incorporated in the Phase 0 contract. The detail of the final contractor WBS should be extended as the program progresses in each phase, to facilitate in-house planning and control.

3.13.1.3 Contract WBS

The contract WBS is the complete WBS applicable to a particular contract or procurement action. It will generally contain the applicable portion of the project summary WBS plus any additional levels of detail necessary for planning and control. The contract WBS outlines program tasks and establishes their relation to the program organization, configuration items, and objectives. It establishes a logical indenture level for correlating performance, technical objectives, schedule, and cost, and ensures that all derivative plans contribute directly to program objectives. It also forms the basis for applying cost and schedule controls, correlating and tracing the contractor WBS to the system requirements, and defining common interfaces between specialty engineering efforts (e.g., technical performance measurement, risk management, logistics engineering, etc.) and programmatic activities (program planning, cost/schedule management, engineering management, etc.). It also plays a key role in ensuring correlation and traceability of WBS product elements.

3.13.1.4 Project WBS

The project WBS is the complete WBS for the program. It contains all WBS elements related to the development and/or production of a Defense item and is formed by combining all the contractor WBSs in a program. The project WBS may be delineated to five or six levels of detail, with the contractor responsible for developing the lower levels identified.

3.13.2 Schedule Planning

Schedule planning involves the preparation of program schedules and includes the development of the program master schedule (PMS) and subordinate schedules, based on the WBS, to ensure that all elements of the contract requirements, including hardware, software, and support items, are delivered on time. Schedules are necessary to integrate the activities of the task organizations to significant milestones.

Schedule planning should commence once the program strategy is confirmed, and requires an understanding of the current project/program dependencies at the time of development. Dependencies include those between engineering activities, those on external activities/organizations, and those by external activities/organizations on engineering products, which may be identified and tracked via either manual or automated techniques, ranging from simple charts to sophisticated activity networks used in PMS production.

3.13.3 Cost and Schedule Control

Cost and schedule control, as described in DoDI 7000.2, *Performance Measurement for Selected Acquisitions*, has two essential objectives that will benefit a major C4I and information systems program. They are: 1) the contractor shall use an effective internal cost and schedule management control system; and 2) the timely and auditable data that the Government can rely on shall be produced by the contractor cost and schedule control system.

The criteria in DoDI 7000.2 ensure that the contractor's management control systems will include policies, procedures, and methods that are designed to provide guidance to the contractor in the areas of organization, planning and budgeting, accounting, analysis and revisions, and access to data. Accordingly, a good management control system includes the following features:

- Measurement of actual work, by the contractor, through "earned value" (i.e., quantifying the amount of planned work that has been accomplished).

- Establishment and control of a program baseline, which represents the contractual schedules and is the cumulative total of all work packages within the contract. Performance is measured against this time-phased budget plan.

- Breakdown of performance measurement by product, through the use of the WBS (i.e., the WBS should completely define the entire program and provide summary levels for performance reporting).

- Breakdown of performance information by organization or function. The cost account is formed at the intersection of the WBS and the contractor's organizational structure. The WBS and functional organization is integrated by identifying the organizations responsible for performing specific tasks.

- Summarizing and reporting of progress information in a disciplined manner. The criteria provides specific formats and data elements that the Government will use to monitor contractor performance, validate contractor status reports, and seek out trends that might affect the program in a positive or negative manner.

- Conduct of variance analysis to identify variances in performance at the cost account level, and corrective action.

3.13.3.1 Cost and Schedule Performance Reporting

Two reports can be generated for the collection of summary contractor performance data. They are: 1) the cost performance report (CPR) and 2) the cost/schedule status report (C/SSR). The reports provide the program manager with contractual information regarding cost, schedule, and technical performance. Both reports are described in DoDI 7000.10, *Contract Cost Performance, Funds Status, and Cost/Schedule Status Reports.* The CPR is used generally to obtain performance data in conjunction with the application of cost/schedule control system criteria (C/SCSC) to a fixed-price incentive or cost-reimbursable contract that meets specified dollar thresholds for research and development or procurement. The C/SSR is intended for the application to contracts more than 12 months in duration where application of the CPR is inappropriate.

The Government can order summary performance data from the contractor's internal control system by placing the requirement for the CPR or C/SSR in the contract (in the SOW and CDRL). In addition to providing an effective channel of communication between the contractor and the Government, the additional benefits of obtaining these data include reporting objective performance status, cost impact of known problems, capability to trace problems to their source (organizational and WBS), and quantification of schedule deviation in dollars from the contract plan.

3.13.3.2 Cost/Schedule Control System

Although many tools on the market, from mainframes to personal computers (PCs), are used for effective program management, no single set of management control systems will meet every contract management data need for performance measurement. Because of variations in organizations, products, and working relationships, it is not feasible to prescribe a universal system for cost and schedule control; however, any system used by the contractor should meet the criteria described in DoDI 7000.2.

The responsibility for developing and applying the specific procedures for complying with the criteria is vested in the contractor. The contractor is required to provide performance data directly from the same system used for internal management control. The basic purpose is to assure that the contractor has in place, and uses, adequate cost and schedule control systems and provides reliable contract status at least monthly.

An element in the evaluation of proposals should be the contractor's system for planning and controlling contract performance. Although DoDI 7000.2 criteria does not require the use of specific systems, the contractor should be contractually required to submit to the program office the CPR and/or C/SSR, at a minimum, on a network system or floppy disk, in a structured American Standard Code for Information Interchange (ASCII) format. The program may in turn use these data to support the many tools available to streamline and automate the analysis and reporting processes associated with analyzing the contractor's reports.

3.14 CONTRACT MANAGEMENT/SOURCE DETERMINATION

The many functions of contract management/source determination are performed by various organizations and individuals, both internal and external to the project/program management office, in the contracting process. This section focuses on those functions and products of the process where the guiding principles for OSE development should be incorporated into the contracting activities and products.

3.14.1 The Request for Proposal

Program managers generally use the competitive proposal method of procurement, in which the RFP is the solicitation instrument. The RFP is a formal, official communication between Government and industry in the contracting process. It describes the Government's needs for goods or services and is the vehicle for soliciting proposals from industry to fulfill those needs. It also provides the frame of reference for source selection, contract definition, and management reviews.

The clarity and coherence with which the RFP is constructed can favorably or unfavorably affect the events to follow. How clearly the Government communicates its need in the RFP, for instance, will almost certainly influence the quality of proposals received, the ease or difficulty in conducting source selection and negotiation, and ultimately, the success or failure of contract performance.

The Federal Acquisition Regulation (FAR) in most cases requires that contracting officers prepare written solicitations and resulting contracts using the uniform contract format outlined in the FAR. The uniform contract format is designed to facilitate preparation of the solicitation and includes Sections A through M, as follows:

- **Section A - Solicitation/Contract Form.** Cover Sheet/Standard Form 33, which contains basic information such as the issuing office address and contract number.

- **Section B - Supplies/Services/Prices/Costs.** Brief description of each contract deliverable (item, quantity, etc.), each covered by a contract line item number. Prices are entered subsequent to solicitation.

- **Section C - Description/Specifications/Work Statement.** Actual tasks to be accomplished in performance of the contract and associated specifications, including the Statement of Work.

- **Section D - Packaging and Marking.** Special packaging and marking requirements such as preservation, protection, and bar coding.

- **Section E - Inspection and Acceptance.** Place of inspection, who will inspect, and acceptance criteria.

- **Section F - Deliveries or Performance.** The time, place, and method of delivery or performance.

- **Section G - Contract Administration Data.** Accounting and paying office information.

- **Section H - Special Contract Requirements.** Requirements unique to the program and the contract (i.e., design to cost, warranties, options, Government-furnished equipment, and incentives).

- **Section I - Contract Clauses.** Commonly referred to as boilerplate and not to be overlooked. Include standard clauses of considerable power defining rights and responsibilities of contracting parties.

- **Section J - List of Attachments.** All attached forms and specifications are listed here, including the CDRL.

- **Section K - Representations, Certifications.** Any special representations required of offerors, such as small/disadvantaged business status, or Equal Employment Opportunity (EEO) compliance.

- **Section L - Instructions, Conditions, Notices to Offerors.** How to organize proposal (volume, page limits, etc.), type of contract contemplated, where to obtain copies of documents, marking of proprietary information.

- **Section M - Evaluation Factors for Award.** How the Government intends to evaluate proposals. These factors are the same as in the Source Selection Plan (SSP), which must be approved before RFP release. Typical factors or evaluation criteria include schedule, management, technical approach, and support.

The principles of OSE and the objectives of the TRM discussed in TAFIM Volume 2 apply across the board in the development of solicitations and are of particular concern in defining the requirements contained in the Statement of Work (Section C). TRM objectives should be understood and the following questions considered in the preparation of the RFP and in source selection:

- Have you specified open standards in your RFP and SOW?

- Have you defined what is expected in conformance and interoperability testing?

- Have you specified a reuse paradigm, reuse repositories, etc.?

- Does the bidder understand Open System issues?

- Is the proposal TAFIM-compliant?

- Has the bidder responded with specific open standards references?

Also, references to Portable Operating System Interface (POSIX) and Federal Information Processing Standard (FIPS) 151-2 should be included in the RFP and SOW as well as requirements specifying adherence to HCI guidelines in order to ensure user portability. (See TAFIM Volume 8, *DoD HCI Style Guide* and use as a reference.) The *Next Generation Computer Resources (NGCR) Acquisition Guide* is a resource that provides guidance and the appropriate wording for inserting Open Systems criteria and requirements into the RFP and SOW.

3.14.2 The Statement of Work

The Statement of Work (SOW) is a mandated requirement of the FAR and is developed by functional managers in the DoD in accordance with MIL-Handbook (HDBK)-245. The SOW is an essential part of the RFP and the heart of the system or equipment procurement. It is also the document by which all nonspecification requirements for contractor efforts are established and defined, either directly or with the use of specifically cited documents. The SOW expresses work efforts as minimal needs and defines those work tasks that cannot be contained in a specification (and must never be included in the CDRL or Data Item Description [DID]); however, it may be supported by specifications or may be used as a supplement to a specification.

The SOW and its associated WBS are the primary instruments upon which contractual costs are based. After the contractor has been selected and the contract awarded, the SOW becomes the standard for measuring the contractor's effectiveness and the basis for change control. As the effort progresses, the Government and contractor refer to the SOW to determine their rights and obligations with regard to contractor responsiveness.

There are five types of SOWs defined for use in MIL-HDBK-245. Four are associated with phases of the life-cycle process. The fifth, for services, is independent of Defense material procurement phases.

3.14.2.1 Type I SOW

This SOW is usually restricted to an expression of goals and objectives when there is a limited ability to accurately identify and define a desired product. Work involving the definition and identification of alternative system design concepts (or a study effort) is usually captured in this SOW type, as are specifications, since typical programs do not have system specifications at this stage of the process.

3.14.2.2 Type II SOW

This SOW type is more descriptive of contractual work efforts and more conclusive in identifying goals and objectives. It is used to refine and define, to a lower level, the details of systems requirements, (development, manufacturing, verification, deployment, operations support, training, and disposal). The Type II SOW is, however, limited in scope to efforts required to proof or prototype, assess results of proofing and prototyping, and define system requirements to the end-item level.

3.14.2.3 Type III SOW

The Type III SOW contains enough detail to enable bidders to translate the program requirements into an effective system SEMP. It also delineates specific tasks for evolving the system requirements and technical objectives into specific system specifications (Type A), which formulate a functional baseline. The Type III SOW is prepared when a specification is used to define the quantitative and qualitative technical requirements for development, manufacturing, verification, deployment, operations support, training, and disposal. Statement of Work tasking would include all those involving the full-scale development and documentation of the intended system.

3.14.2.4 Type IV SOW

This SOW is used to culminate end efforts of the development phases by supporting production and ultimate deployment of the system. Typical tasks include producing and deploying the system per specifications and approved engineering changes, providing interim support, performing sustaining engineering and configuration management, and developing and delivering logistics support.

3.14.2.5 Type V SOW

The Type V SOW is used when the need for contractor support is identified independent of the actual development and procurement of the C4I and information systems. (Please refer to MIL-HDBK-245 for more detailed information and guidelines regarding the SOW types and SOW preparation.)

3.14.3 Selection of Standards and Specifications

Every DoD program has a set of unique specifications that define its specific technical requirements. These documents incorporate or refer to many Government standards to define items, approaches, or procedures that may be used in the development and production process. These Government standards are employed to give new programs the benefit of previous technical experience, to promote interchangeability and commonality, and to minimize costs of ownership. Implementation must be carefully considered to ensure that general standards/specifications represent current technology, yet do not create unnecessary costs to the program.

3.14.3.1 Specification and Standards Categories

Specifications are documents prepared to support acquisitions and to describe items that vary greatly in complexity. Specifications form the skeleton around which the Defense LCM process is built and are necessary to satisfy the primary objective of any procurement action. Specifications will establish the requirements in terms of both design detail and performance. There are two basic categories of specifications: general specifications, and program peculiar specifications. General specifications, referred to as military specifications, are controlled by the Defense Standardization and Specification Program (DSSP) and apply to all acquisition programs. These specifications represent a particular requirement at a particular time that can be used over and over again on many different programs. They include specifications for materials, parts, and processes; test criteria documentation; and management specifications.

Program peculiar specifications apply only to those products developed to meet specific operational requirements. The basic forms and types of these specifications are defined in MIL-STD-490A and include the system/segment specification, development specification, product specification, process specification, and material specification. As described in Section 3.5, standards are documents that establish engineering and technical requirements for processes, procedures, practices, and methods that have been adopted unilaterally.

The order of precedence for specifications and standards is (highest to lowest): Specifications (Federal, military, program peculiar); Standards (federal, military, industry); and Handbooks (Governmental). Procedures and policy for the DoD Standardization and Specification Program are promulgated by DoDD 4120.3. Specifications, standards, handbooks, and other engineering documentation prepared under DSSP are intended to state only the actual needs of the Government in a manner that will encourage maximum competition. The objectives of the DSSP are contained in DoD 4120.3-M, *Defense Standardization and Specification Program Policies, Procedures, and Instructions*, of August 1978.

3.14.3.2 Specification and Standard Selection

Government and industry are jointly responsible for ensuring that each specification and standard imposed on a contract is suitably tailored and current. The AITS in TAFIM Volume 7 should be used in selecting specifications and standards, as well as the ITSG discussed in Section 3.5. The ITSG provides amplifying implementation guidance for those standards identified in TAFIM Volume 7 and supporting information on AITS standards hierarchies.

3.14.3.3 Streamlining and Tailoring Methods

The objective of streamlining and tailoring is to clearly communicate what is required in functional performance-oriented terms at the beginning of development, and to allow flexibility for the application of the contractor's experience and judgment. Once specifications and standards have been selected for a program, it is necessary to review and tailor the requirements contained in each specification and standard before RFP release, as well as at each milestone in the program life-cycle, if necessary. There are a number of ways to tailor specifications and procurement standards. For example, the application of a standard may be limited to specified components, or types of components, within the system by specifying the limits in the body of the system specification. Applicable portions of a standard may also be extracted for incorporation into the text of a development specification. In either case, a referenced standard may be supplemented by descriptive text in the specification to clarify the intended requirements or application. Inapplicable portions of the standard may be deleted by identifying them in an appendix to either specification.

The following are rules of thumb for specification and standards tailoring:

- At Milestone 0, specify system-level requirements in mission performance terms. Before full-scale development, military specifications and standards should be cited for guidance only.

- For development contracts, contractual applicability of specifications, standards, and related documents should be limited to those cited in the contract, and to specified portions of documents directly referenced by those cited (first-tier references). All other referenced documents (second-tier and below) should be for guidance only, unless specifically called out in the contract.

- For production contracts, those specifications, standards, and referenced documents comprising the baseline for production should be considered contractual requirements for procurement and re-procurement purposes. Acquisition streamlining should continue throughout the production phase, with emphasis on ensuring that only essential production and data requirements are carried forward into follow-on production contracts.

- When a decision is made to use COTS/NDI, all specifications and standards that define the product/items should be contractually specified in the solicitation.

- During the design process, the contractor should be required by contract to recommend detailed specifications, standards, and requirements to be applied as the system evolves toward the end product. For instance, as the system design evolves through Phase I, lower-tier specifications and standards should be selected and tailored for the next phase. Also, identified requirements should be reviewed by systems engineering; tailored, as appropriate; and identified as requirements in the development proposal. During development, a primary task should be to review and scrub lower-tier references to ensure that those specifications and standards are cost-effective. The program manager should make the final determination as to which data requirements statements, specifications, and standards should apply in production (Phase III) and throughout the remainder of the program.

Additional guidance on streamlining and tailoring is included in DoDD 5000.43 and DoD-HDBK-248, which specifies the use of contractor's management systems, internal procedures, data formats, etc., unless the program office determines that these do not meet program needs. This increased emphasis on contractor systems, procedures, and documentation increases the contractor's flexibility in generating program documentation in the most efficient and effective manner. DoDD 5000.43 further specifies procedures regarding the contractual referencing aspects of the streamlining initiative, which calls for practical measures to preclude untimely, untailored, and accidentally referenced application of military specifications and standards; that is, to specify required results rather than detailed how-to procedures in RFPs and contracts.

3.14.4 The Contract Data Requirements List

The CDRL (DD Form 1423) is the mechanism for ordering and delivering recorded information, regardless of medium or characteristics, of any nature, including administrative, financial, and technical. Several rules govern the contractual acquisition of data. Data must be set forth in a contract in a very specific way if the contract is more than $25,000. (Data requirements may be specified in the specifications/SOW if the overall contract is estimated to be less than $25,000.) With the exception of data specifically required by the FAR or Defense Federal Acquisition Regulations (DFARS), or specifically exempted by the DFARS, all deliverable data must be listed in the CDRL.

The CDRL provides a single place in the contract for directing the contractor to prepare and deliver data and to meet specific approval and acceptance criteria. It establishes data required, delivery characteristics, the degree of tailoring to be applied to the DID, the points for inspection and acceptance, any interim approval requirements, and the price of the data, by DID.

Data format and content are established by data acquisition documents (usually DIDs), which, with the exception of one-time DIDs, are approved and given OMB clearance by the Defense Quality and Standardization Office. DIDs (DD Form 1664) define the data required for delivery by the contractor, including content and preparation instructions, format, intended use, and other source documents that may be used to describe the data to be delivered.

DoD 5000.19-L, *Acquisition Management Systems and Data Requirements List (AMSDL)* lists all the data acquisition documents (with the exception of one-time DIDs) that are approved and given OMB clearance in accordance with Part IX, Section B, of DoDI 5000.2. Part I of the AMSDL lists source documents and related DIDs by data functional area assignment. Part II is a numerical listing; Part III lists DIDs by key word; and Part IV lists canceled and superseded source documents and DIDs.

The DISA *Acquisition How-To Guide* (Chapter 9, "Explanation of Forms"), accessible through the DISA Library, is an excellent source for obtaining additional information on DID selection and CDRL development.

3.14.5 Source Selection Procedures

The primary objectives of the source selection process are to: (1) select contractors who can best meet Government needs as described in the solicitation/RFP; and (2) ensure that the source selection process provides for the impartial, equitable, and comprehensive evaluation of each offeror's proposal and minimizes the cost of the selection process to the Government and industry. The source selection process is managed by a three-level organization or team composed of the Source Selection Authority (SSA), the Source Selection Advisory Council (SSAC), and the Source Selection Evaluation Board (SSEB). The procedures for source selection are contained in the SSP, which the program manager prepares. The remainder of this section addresses the roles and responsibilities of the source selection team and the purpose and content of the SSP. Additional information on source selection can be found in the FAR, Subpart 15.6, "Source Selection"; DoD Instruction 5000.2, Part 10, Section B; Air Force Regulation (AFR) 70-15, "Proposal Evaluation and Source Selection"; Army Regulation (AR) 715-6, "Proposal Evaluation and Source Selection"; and Secretary of the Navy Instruction (SECNAVINST) 4200.33, "Selection of Contractual Sources for Major Defense Systems."

3.14.5.1 Source Selection Authority

The Source Selection Authority (SSA) is the Service Secretary/Component head for major systems, responsible for the overall source selection activity, but authority may be delegated to the next level. Responsibility includes approval of the Source Selection Plan, establishing the membership of the SSAC, and making the final selection decision. The SSA also ensures the evaluation criteria are consistent with the solicitation and policy.

3.14.5.2 Source Selection Advisory Council

The Source Selection Advisory Council (SSAC) is a group of senior military and/or civilian personnel representing various functional and technical disciplines. The SSAC is responsible for appointing the membership of the SSEB, establishing and applying the evaluation criteria and the numerical weighting (scoring scheme) for these criteria. The SSAC also reviews the SSEB findings, prepares an analysis of each offeror's proposal, and compares the proposals to one another. The SSAC, unless a performance risk assessment group is employed, is the body that considers contractor past performance. The output of the SSAC is a final report to the SSA on SSAC evaluations.

3.14.5.3 Source Selection Evaluation Board

The SSEB is composed of military and/or civilian personnel representing a variety of functional and technical disciplines and is assigned by the SSAC to evaluate proposals and provide narrative findings to the SSAC for use in its review. The leadership of the SSEB should be of importance to the program manager, since the staffing would consist of a cross-section of expertise from within and outside the organization, which typically includes personnel from logistics, cost analysis, operational, contract, legal, and technical areas.

3.14.5.4 The Source Selection Plan

The Source Selection Plan (SSP) establishes procedures for accomplishing the above-mentioned prime objectives. Before a solicitation is issued, the SSA approves the SSP. The program manager is responsible for preparing the plan and obtaining SSA approval before releasing the solicitation. The plan summarizes the overall acquisition strategy contemplated for the requirement and includes a discussion of the extent of competition expected, a description of the evaluation techniques to be used, and the schedule of significant actions required. It also describes the organization, membership, and responsibilities of the source selection team and identifies the evaluation factors and detailed evaluation procedures, which mirror section M of the RFP. The specific evaluation criteria are listed in the order of their importance and may include technical aspects, operational considerations, supportability management capabilities, and cost analysis. Past performance may be also be considered as an area or as an item. Representative examples of the items considered in each of these evaluation criteria areas include:

- **Technical**
  - Design Approach
  - Test Plan
  - Performance Criteria
  - Design Innovation

- **Operational**
  - Approach to Operational Concept
  - Maintainability
  - System Capability
- **Supportability**
  - Impact on Current Logistics Systems
  - Maintenance Concept
  - Supply Support
- **Management**
  - Integration Procedures
  - Interface Procedures
  - Schedule Adherence
  - Program Control
  - Past Performance
- **Cost**
  - Risk
  - Interface Procedures
  - Labor and Overhead Rates
  - Development Costs
  - Life-Cycle Costs
  - Cost Realism.

3.14.6 The Technical Data Package

The Technical Data Package (TDP) is a technical description of an item adequate for use in procurement. This description defines the required design configuration and assures adequacy of item performance. It consists of all available data such as plans, drawings, and associated lists, specifications, standards, models, performance requirements, quality assurance provisions, and packaging data, and may range from a single line in a contract to several hundreds or thousands of pages of documents. It does not include computer software or financial, administrative, cost or pricing, or management data, or other information incidental to contract administration.

The guiding standard for the TDP is MIL-T-31000, which prescribes the requirements for potential data elements and data management products for inclusion in the TDP. These requirements are tailored by the Government for inclusion in the CDRL of the solicitation/RFP, and may be tailored by the contractor in response to a solicitation using the guidelines of MIL-HDBK-248.

Contract provisions should ensure that contractors and subcontractors prepare and update TDPs as an integral part of their design, development, and production efforts. Technical data (and technical manuals) should be updated to reflect approved design changes to be made available concurrent with the implementation of the change. Additionally, the TDP that the contractor delivers to the Government should be representative of the product baseline and should have sufficient detail to permit duplicate fabrication by any competent commercial source without additional investment in design or development. However, experience indicates potential errors, omissions, inaccuracies, or nondisclosures in a TDP may pose cost, technical, and schedule risks if used in follow-on contracts; thus, TDP validation is necessary to mitigate this risk.

TDP validation should be a controlled process by which technical data can be certified as acceptable for intended use. The best validation method for use on a C4I and information systems program is the Functional and Physical Configuration Audit (see MIL-STD-973) of the producer's TDP to ensure the accuracy of drawings and other technical and supporting documentation against the design and in accordance with prescribed specifications and standards.

3.15 SYSTEMS ENGINEERING/TECHNICAL MANAGEMENT

In simple terms, systems engineering is both a technical process and a management process. The following definition identifies the technical side to systems engineering:

The application of scientific and engineering efforts to (a) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test, and evaluation; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; (c) integrate reliability, maintainability, safety, survivability, human engineering, and other such factors into the total engineering effort to meet cost, schedule, supportability, and technical performance objectives.

Another popular definition favors the management approach and defines systems engineering as:

The management function which controls the total system development effort for the purpose of achieving an optimum balance of all system elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize the overall system effectiveness.

With respect to each of these definitions, both the technical and management aspects of systems engineering should be applied throughout the system life-cycle to produce a successful operational system. In the planning stages of the system life-cycle, systems engineering is essential in conceiving the system concept, establishing architectures, and defining known and implied user requirements. As the detailed design is being done, systems engineers assure a balanced influence of all required design specialties, resolve interface problems, conduct design reviews, perform trade-off analyses, and assist in verifying system performance. During the development phase, concern is with verifying requirements compliance and system capability, maintaining the system baseline, and forming an analytical framework for producibility analysis. During system operations and support, systems engineering evaluates proposed changes to the system, establishes change effectiveness, and facilitates the incorporation of change modifications and updates.

The major technical tasks and the primary application of the systems engineering process are accomplished by the contractor. The quality of effort by the contractor is largely dependent on a well-defined contract that defines the Government/industry agreement with respect to the system under consideration (see Section 3.14). The RFP sets forth the systems engineering needs; the SOW provides the formal statement of those needs as requirements for the contractor; the "specification" defines the technical system requirements; and the CDRL identifies data deliverable requirements.

3.15.1 The Systems Engineering Process

Although programs differ in underlying requirements, the systems engineering process offers a consistent, logical process for accomplishing system design tasks. The process itself leads to a well-defined, completely documented, and optimally balanced system with a complete set of documentation tailored to the needs of a specific program. Figure 3-3 illustrates the interactive activities of a basic systems engineering process. This process may be iterative and recurring during each life-cycle phase and whenever a change is initiated or needed to provide the progressive definition of the system, subsystem, and configuration items, and their verification. The level of detail involved should be commensurate with the contractual objectives of the program.

The major elements of systems engineering, including the activities and outputs of the systems engineering process, are summarized in Appendix E.

3.16 SOFTWARE ACQUISITION MANAGEMENT

Software acquisition management is the process of acquiring software, managing its development, and ensuring its supportability for the entire life-cycle. Software acquisition management activities include planning, contracting, budgeting, evaluating performance, and providing for future support of the system, as well as acquiring software, usually by contract, from a third party. Typically, the three organizations involved in the process include the customer or user of the system, the contracting agency or buyer, and the developer or seller. Depending on the scope of the effort, there may possibly be many agencies and contractors involved. While software engineering concentrates on building the software, project management focuses on managing the engineering development or acquisition.

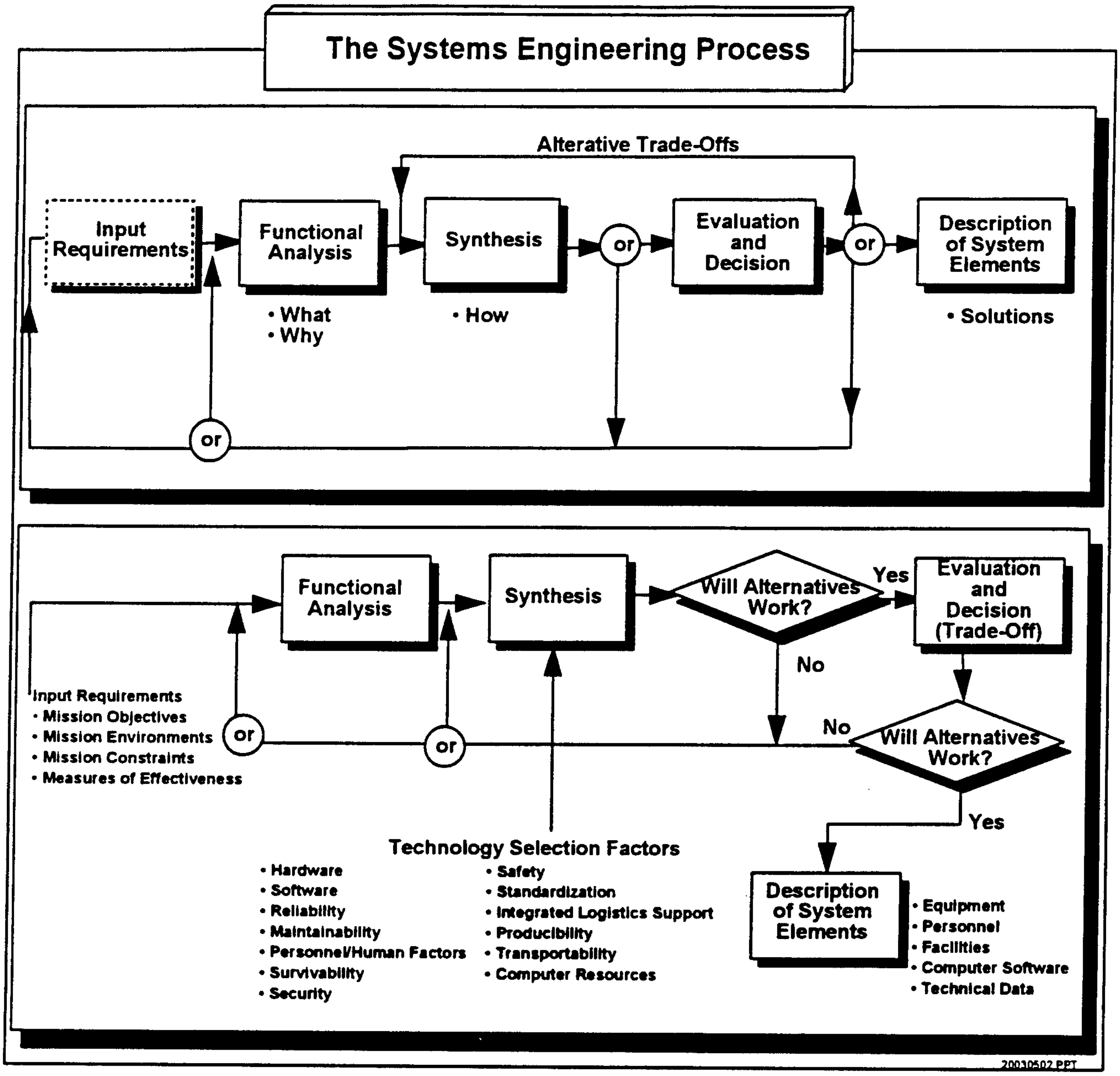


Figure 3-3. The Systems Engineering Process

The acquisition of software commonly follows the LCM process depicted in the DoD 8120 series directives. During concept exploration and definition (Phase 0), the buyer develops requirements, prepares specifications, and develops an acquisition strategy. During source selection, a vendor or developer is chosen to develop the system, based on the proposal made by the vendor or developer. During demonstration and validation (Phase I) and throughout the remainder of the contract period, the vendor's or developer's progress and compliance with contract provisions are monitored.

3.16.1 Planning the Acquisition

Software acquisition planning begins when the requirements start to be prepared (see Sections 3.3, 3.11, and 3.15.1). Because of the lead times involved in competitive procurement, the buyer and seller resources must be put into place well in advance of the contract. The program manager, once in place, is also well advised to immediately begin planning the acquisition and development activities for the remainder of the LCM process. There are two key planning documents in any software acquisition: the PMP and the SDP. The PMP is prepared by the Government and sets the tone for the entire acquisition/development, whereas the SDP, prepared by the contractor, focuses on software methods, tools, and resource issues, and provides the detailed information on how the software will be developed. The key considerations that the PMP and SDP should address include organization and interfaces, activity structure, schedule and milestones, resources, support, subcontractor management, software methodology, reviews, documentation, software environment, testing, product evaluations, and risk management.

The primary planning tool is the WBS (see Section 3.13.1), which should be outlined in the RFP (see Section 3.14.1). Once the WBS has been defined, each of the tasks identified within it can be scheduled, and resources can be estimated.

3.16.2 Life-Cycle Standards

The mechanism used to structure the software acquisition process (including software development) and define the major activities associated with it is the life-cycle model selected for the acquisition. The life-cycle model is a process model and mechanism for communicating to the managerial, technical, and user personnel associated with the program or project what work tasks need to be accomplished, when, and by whom. The most widely used life-cycle process model for software development is the waterfall life-cycle model. While advanced models may be used to structure the work in complex software developments (e.g., the spiral model may be used to incorporate prototyping as a risk reduction option at any stage), the waterfall model can be used to communicate the sequence of events and work that must be accomplished to develop a software product. This model has been institutionalized in a number of standards that provide a basis for management, thus supplying an acquisition infrastructure for the program or project. These standards are among the popular sources of life-cycle process standards contained in TAFIM Volume 7, Appendix A, "Adopted Information Technology Standards (AITS) Table" and in the AITS companion document, the *Information Technology Standards Guidance (ITSG)*.

MIL-STD-498, *Software Development and Documentation*, is the most widely used standard for software development and life-cycle management. It is a management and engineering standard that sets forth requirements for software development and prescribes a uniform software development process. It contains requirements for software development management, software engineering, configuration management, product evaluation, formal qualification testing, transitioning software to the operational environment, and content and format requirements (DIDS) for software data deliverables, the documentation that establishes the baselines to be used to control system design and development. As with all standards selected for a program, tailoring of this standard is recommended (see Sections 3.5 and 3.14.3).

3.16.3 Software Management Environment

The program organization responsible for the management of software development or acquisition should be a highly visible part of the program structure and high enough in the organizational hierarchy to command the resources necessary to do its job effectively. Lines of communication in the program should be structured to expedite vertical as well as horizontal flows. Cross-functional teams also aid in problem resolution involving cross-organizational boundaries. Working groups also aid in problem resolution. Plans to change the organizational structure as the program moves from definition through testing to operations should also be made, so that the right resources are available to perform and support planned activities in each life-cycle phase.

An adequate software environment is also required in both developer and customer organizations. A software environment consists of the set of hardware, software, and firmware used to perform the development effort. Typical elements of the environment include equipment (workstations, file servers, communications networks, etc.), assemblers, compilers, database managers, debuggers, editors, library systems, simulators, CASE tools, and a variety of other tools. Communications are enhanced when both the development organization and the customer have access to the same information stored within the environment.

3.17 INTERFACE MANAGEMENT

Interface definition, management, and control are integral parts of the systems engineering and configuration management processes. Systems engineering is concerned with the identification, documentation, and management of all functional and technical interfaces of a system, its components, support equipment, operating/applications software, and facilities. Interface control is achieved through the CM process as interface requirements are baselined, proposed, and changed. Interface management of an Open System will most likely involve the acquisition of hardware and the development of software applications that will interface with other systems and subsystems. This will require effective interface management to be implemented in the systems engineering and CM processes, to identify and document interfaces, ensure hardware/software standardization, resolve interface problems, and adhere to functional/technical interface requirements. Interface management should be implemented in accordance with the configuration management plan of the program and any and all agreements made between the interfacing parties to ensure interfaces are identified and documented in system design documentation and controlled during system development and operations.

3.17.1 Interface Types

An interface, as defined in MIL-STD-973, is "the functional and physical characteristics required to exist at a common boundary." In other words, an interface is "identified" when a common boundary exists between two system entities. It is "defined" when characteristics are completely specified (i.e., functional, physical, protocol, performance, data source/destination, frequency/timing levels, data format/content/rate/volume, security characteristics, etc.). The following are the types of interfaces that are typically controlled in an OSE:

- **External interface.** An interface that exists between hardware, software, or both, where design and/or in-service support responsibilities for the two sides of the interface are under the control of different DoD and/or DoD Component activities.
- **Internal interface.** An interface that exists between hardware, software, or both, where design and/or in-service support responsibilities for the two sides of the interface are under the control of the same DoD Component activity and may involve different contractors.
- **Single-entity interface**. An interface that exists between hardware, software, or both, where design and/or in-service support responsibilities for the two sides of the interface are under the control of the same DOD Component activity and the same contractor.

3.17.2 Interface Requirements

Interface requirements must be included in system and development specifications. The development specifications may further allocate interface requirements to lower-level Components, where these requirements will be functionally and physically met. System interface agreements (SIAs) (or other documents deemed as interface control documentation [ICD] for a program) are typically developed for each system application in order to depict the functional and physical interfaces of related or co-functioning items. The SIA/ICD provides the means to measure, evaluate, and formally control the record layout/structure of system data transmissions and record interface agreements between functional areas. The SIA/ICD also serves as the primary document for system interface control and becomes part of the program's technical baseline. A separate SIA/ICD should be developed for each automated interface and updated as a living document throughout the applications life-cycle.

3.17.3 Interface Control

The program's systems engineering management organization and the designer/developer/integrator of the system are jointly responsible for the identification and control of the system's external, internal, and single-entity interfaces. This joint responsibility may be managed through the SIAs/ICDs described above, and by the establishment of an Interface Control Working Group (ICWG), a recommended mechanism for ensuring interface control. The ICWG typically consists of Government and contractor representatives, and representatives from the respective functional areas interfacing with the system at hand. The role of the ICWG is to resolve interface management issues and assess and determine data transfer requirements, including the data needed to meet those requirements. The ICWG normally performs interface management and control tasks from Milestone I to Milestone III.

3.17.3.1 Interface Change Control

Changes to a system application and/or interfacing system during development, testing, or implementation that affect the communications link between organizations or other interface-related issues are typically handled through the program's configuration management organization. Changes and related issues include procedural modifications, hardware or software changes, data element standardization changes, changes to editing criteria, input or output format changes, and frequency of use deviations. The organization assigned as the technical lead for a configuration against which a proposed change is issued ensures interface impact and potential related change analysis through the ICWG. The ICWG determines that interface change requirements have been properly assessed and documented in related change documentation before the technical lead organization approves the basic change. The requirements for the identification, documentation, and coordination of related engineering changes are further defined in MIL-STD-973 (Section 5.4.2.3.6 and Section 6).

3.18 TEST AND EVALUATION

Test and Evaluation (T&E) is an iterative process of measurement, analysis or feedback, corrective action, and retest. It is used throughout the LCM process to reduce technical and program risk and to provide early and continuing estimates of the system's operational effectiveness and suitability. Issues and criteria are developed from operational requirements and performance thresholds and objectives found in early program documents, such as the MNS, program baseline, and requirements documents. Test methods and measurement include data collection (including field test, test beds, and simulations) designed to evaluate the conformance of system components to standards of performance. From a systems engineering perspective, test planning, testing, and analysis of test results are integral parts of the basic systems engineering process. T&E encompasses relationships with all system elements, such as equipment, software, facilities, personnel, and procedural data.

The successful accomplishment of T&E objectives is a key requirement for milestone decisions to commit additional resources to a program or to advance the program from one life-cycle phase to the next. In this respect, test planning needs to be initiated early in the LCM process so that appropriate test activities can be fully integrated into the overall development process.

T&E programs for C4I and information systems fall under the responsibility of the DoD Director, Test and Evaluation (D, T&E) and DoD Director, Operational Test and Evaluation (D, OT&E). Both organizations coordinate and develop and maintain DoD-level T&E policies, procedures, and other guidance by which C4I and information system test programs are assessed and validated through the milestone review process. T&E policy and procedures, described in DoDD 8120.1 and 8120.2 direct the establishment of a T&E program in accordance with the DoD 5000 series directives, in particular DoDI 5000.2, which further identifies the responsibilities for test program oversight, the requirements and guidelines for Developmental Test & Evaluation (DT&E) and OT&E, the major categories of T&E to be implemented. Additionally, DoD 8120.2-M, Part 7, provides procedures and formats for preparing the TEMP, which documents the overall structure and objectives of the T&E program. A brief overview of the TEMP and the functions of DT&E and OT&E follow in the subparagraphs below.

3.18.1 Test and Evaluation Master Plan

The Test and Evaluation Master Plan (TEMP) is a broad, top-level plan detailing all major T&E events and is a primary document used in the LCM review and decision-making process. The TEMP covers the program life-cycle from initiation through post deployment, including major modifications or upgrades, and defines how the system components will accomplish the planned testing and evaluation for each life-cycle phase in order to support major program decisions. It identifies special T&E resources and requirements to facilitate long-range planning, including the cost of contracted telecommunications, training, Automated Data Processing (ADP), and consulting services; documents major agreements between the material developer and the independent operational T&E agent, and includes the rationale and schedule for planned tests. It also relates the T&E effort clearly to technical characteristics, technical risk, operational issues and concepts, system performance, reliability, availability, maintainability, logistics requirements, and major decision points. A program's first, preliminary TEMP is submitted in support of the Milestone I decision. TEMP updates are then required before each subsequent decision milestone. Additional updates are required when the program baseline is breached or when the program has changed significantly.

The DoD guidelines for TEMP coordination and approval are contained in DoDD 8120.1, DoDI 8120.2, and DoD 5000.2. TEMP preparation is in accordance with the required and specified format of DoD 8120.2-M, Part 7. For multi-service or joint programs, a single, integrated TEMP is required, with requirements unique to a DoD Component annexed to the basic TEMP. For Multi-system programs, a Capstone TEMP integrating the T&E program for the entire system is prepared.

3.18.2 Developmental Test and Evaluation

The Developmental Test and Evaluation (DT&E) is conducted throughout the LCM process to ensure the acquisition and fielding of an effective and supportable system. DT&E is normally planned, conducted, and monitored by the developing agency (joint responsibility of the program manager and contractor) to:

- Assist the design and development process
- Verify performance objectives and specifications
- Demonstrate that design risks have been minimized
- Estimate the system's utility
- Provide assurance that the system/equipment/component is ready for testing in the operational environment.

DT&E includes the T&E of components and subsystems at all WBS levels, including hardware/software integration, related software testing, and production acceptance testing. It emphasizes the use of controlled conditions and well-trained operators and maintainers, and may involve the use of simulations, models, test beds, full-scale engineering development models, and prototypes of system components or the system itself. DT&E can include conformance testing, which includes testing products to the requirements of an Open System interface standard developed through, and approved by, independent standards bodies (i.e., National Institute of Standards and Technology[NIST], ISO, IEEE, ANSI); interoperability testing, which involves the testing of two or more interface-connected products for their ability to work together; and performance testing, which includes the verification of interface performance criteria. While its goal is to verify the attainment of technical performance specifications and objectives, feedback from DT&E results provides meaningful input to risk assessment decision-making.

DT&E is conducted during the concept exploration and definition phase (Phase 0), to assist in selecting preferred alternative system concepts, technologies, and designs. During the demonstration and validation phase (Phase I), DT&E is conducted to identify and validate the preferred technical approach, including the identification of technical risks and feasible solutions. During development (Phase II), DT&E should demonstrate that engineering is reasonably complete, that all significant design problems have been identified with solutions in hand, and that the design meets the required specifications in all areas, such as performance, reliability, and maintainability, within the range of parameters specified for operational deployment. After the Milestone III decision (production and deployment, Phase III), DT&E is an integral part of the development, validation, and introduction of system changes undertaken to improve the system, to react to new requirements, or to reduce life-cycle costs.

3.18.3 Operational Test and Evaluation

For major systems, Operational Test and Evaluation (OT&E) is typically conducted by a major OT&E field agency located within the DoD Component. This operational test agency (OTA) must be separate and independent from both the developing/procuring agency and the using agency. The OTA is responsible for managing operational testing, reporting test results, and providing its independent evaluation of the system being tested to the Military Service Chief or Defense Agency Director for Operational Test and Evaluation, who will approve the organizational structure of the OTA. The principal objectives of OT&E are to:

- Estimate the operational effectiveness and operational suitability of the system
- Identify needed modifications or improvements
- Provide information on tactics, doctrine, organization, and personnel requirements
- Provide data to uphold or verify the adequacy of various manuals, handbooks, supporting plans, and documentation.

OT&E is planned and conducted in an environment as realistic as possible, and can be combined with DT&E when significant, clearly identified cost and schedule benefits will result. Typical operation and support personnel should be used to obtain a valid estimate of the user's capability to operate and maintain the system when deployed; however, the contractor is precluded by public law from participating in realistic OT&E. Operational testing is conducted during the concept exploration and definition phase (Phase 0) to estimate the operational impact of candidate technical approaches and to assist in selecting alternative preferred concepts; during the demonstration and validation phase (Phase I), to examine the operational aspects of the selected alternatives, estimate the potential operational effectiveness and suitability of the candidate system, and identify operational issues for early assessment and future operational testing; during development (Phase II), to demonstrate the system's operational effectiveness and suitability; and after the Milestone III decision (production and deployment, Phase III), to test the fixes to be incorporated into the production or deployment system and to validate the achievement of program objectives.

Although OT&E is planned and conducted by an independent testing activity, the program manager must closely coordinate all aspects of test and evaluation with the OTA, to ensure that DT&E objectives coincide with OT&E objectives.

3.19 LOGISTICS MANAGEMENT

Integrated Logistics Support (ILS) is defined as a composite of the elements necessary to assure the effective and economical support of a system or equipment at all levels of maintenance for its programmed life-cycle. It integrates logistics support elements into complementary time-phased and mission-oriented actions to plan, develop, acquire, and operate equipment. It is implemented as a disciplined, unified, and iterative approach and process to the management and technical activities necessary to integrate support considerations into system and equipment design; develop support requirements; acquire the required support; and provide the required support during operations, at minimal cost. As with other conventional acquisition approaches, ILS is critical to C4I and information system acquisitions, in order to ensure that system design is influenced by support requirements and that support is available for operational sustainment.

The program manager establishes an ILS program in accordance with the requirements of DoDD 5000.2, Part 7, Section A, and may include such ILS areas as logistics support analysis (LSA) and Planning (in accordance with MIL-STD-1388-1B); reliability, availability, and maintainability; supply support, test, and support Equipment; transportation and handling; personnel and training; facilities; technical data and publications; post-production support; and the development of ILS documentation such as the ILSP, Logistics Support Analysis Records (LSAR) (in accordance with MIL-STD-1388-2B), and the Deployment Plan. The overall foundation and objectives of the ILS program are contained in the ILSP, which is developed in accordance with DoD 8120.2-M, Part 13.

3.19.1 Integrated Logistics Support Plan

The Integrated Logistics Support Plan (ILSP) is a management tool that delineates anticipated future logistical planning actions by the program office and external supporting activities. Its function is to identify what logistics support tasks will be accomplished, how and when they will be accomplished, and who will be responsible for their accomplishment. The ILSP is considered the foundation document for coordinating logistics planning efforts to ensure that each of the ILS elements is addressed and integrated with the other program elements throughout the life-cycle. It contains the details that form the basis for specific actions by supporting activities and for developing logistics requirements to be included in contractual documents. The ILSP provides for coordinated actions on the part of logistic element managers and the contractor, and it documents the manner in which each logistic support element is to be obtained, integrated, and sustained.

The program manager is responsible for initiating the ILSP at the outset of the program, in the concept exploration and evaluation phase (Phase 0). The content and format may vary according to Service and should be subject to tailoring, based on program nature and needs. The planning should be focused to the subsystem level and should include the coordination and input of all required and participating staff agencies. When approved, the ILSP becomes the implementation plan for all participating activities and is treated as an integral part of the Program Management Plan. The ILSP should be updated when new program direction is received, when changes involving personnel, training, facilities, and other ILS elements occur, and when there are major system configuration changes.

3.20 METRICS

The increasing complexity of DoD systems, the need for evolutionary or incremental developments, and the migration of legacy systems have traditionally made program management and development a difficult task in itself. Overlaying additional requirements (i.e., imposition of reuse, new development methodologies, languages, processes, and environments) on top of these life-cycle elements further complicates a manager's role and responsibilities. Furthermore, new demands created by complex mission support activities, cross-functional interfaces, Open System requirements, and standards are added burdens to a manager's sphere of operation and influence. Thus, the issue of quantification through metrics application (i.e., understanding what to measure and collect and when to collect it), becomes a significant task in light of the extensive and multiphased life-cycles that drive a particular system development.

A metric is a quantitative value or set of values derived from measurement data that provides an indication of progress, product quality, or resource utilization. Measurement data is quantitative data that directly characterizes some aspect of a project. Metrics application is an important means of monitoring and evaluating the progress of any work effort. Proper use of metrics data can help to manage development, mitigate risks, control costs, and avoid problems.

The various types of metrics that may be employed in a program are briefly discussed in the sections that follow. A more extensive discussion of metrics and their effective use can be found in the following publications: *Practical Software Measurement, DoD Software Performance Engineering (SPE) Project, Software and Performance Metrics Assessment.*

3.20.1 Reuse Metrics

The many variations and deviations of the particular acquisition and development paradigm can easily alter the sequence of events (e.g., design reviews), and the type of information needed for an event or milestone activity (i.e., Milestone I, II, III, or IV). Development under a reuse paradigm requires an earlier review of specific software and design elements, by virtue of their existence, to establish feasibility of the identified reusable software component. It is in the best interests of the program manager and DoD to have a set of measures and metrics on a particular reusable element attesting to its integrity, reliability, and liabilities. The same concept of prior knowledge, quantification, or assessment applies to a contractor selected for the system development in terms of the contractor's ability to develop software of a certain complexity or size. The same argument can be made for the development processes to be encountered, their stability, and their maturity.

3.20.2 Requirements Metrics

Requirements and their related issues and maturity exist in the systems, software, and hardware phases of the life-cycle. Their traceability is of concern to systems, software, and hardware engineers. The collection of requirements metrics should be similar and defined in a consistent manner. Thus, program managers should be aware of the potential for instrumentation across more extensive life-cycle activities and domains, and should focus on common denominators across these disciplines. Systems requirements decompose into lower-level ones, giving rise to allocated and derived requirements. As requirements mature and stabilize, their numbers increase by orders of magnitude and are dispersed across a system's documentation. Requirements expansion and categorization has been recognized in standards for many years. How to group and associate lower-level requirements into effective testing sets that can subseqùently be combined into a minimum set of larger system test sets has always been a difficult issue. These same issues are found across domains (e.g., software, systems, hardware). Requirements maturity, stability, traceability, and testability characteristics have also been difficult to capture in supporting design automation and CASE tools. Focusing on requirements common denominators and their metrics across these domains would be of significant consequence to program managers. Changes in requirements are indicative of changes in scope, resulting in a corresponding cost and schedule impact. An awareness of these common denominators enables the program manager to collect metrics earlier in the life-cycle in a more consistent manner. The ability to collect metrics earlier thus provides for better risk mitigation, effective problem resolution, and cost avoidance. Since the identification of common software and systems engineering metrics is now possible, a more uniform collection, traceability, and analysis of these metrics and a definition of viable metrics programs can be obtained.

3.20.3 Migration Metrics

Migration metrics are becoming increasingly important, since the number of legacy systems being transitioned or updated by DoD is increasing. The migration of systems is expected to continue, since DoD resources to build new systems are scarce. Migration of legacy systems becomes even more important in the face of inter-Service operational and cross-functional demands and the need for greater interoperability and use of open standards.

3.20.4 Software Metrics

Software performance metrics are worthwhile and should begin to be incorporated into a software projects metrics program from cradle to grave. These metrics can have a significant impact on the design of software systems when software performance models are applied in the concept and requirements phases. Projecting performance requirements may warrant complete design changes before costly implementation.

Six common metrics have been identified for SPE:

- Response Time
- Throughput
- Workload Specs
- Resource Usage
- Transaction Frequency
- Capacity.

These metrics are the most useful and should be used throughout the system life-cycle process. Estimates should be provided in the concept exploration and evaluation through development phases, and actual measurements should be taken during implementation, test, integration, and operations and maintenance.

3.21 REUSE

Reuse simply means "to put or bring into action or service again or to employ for or apply to a given purpose again." When properly planned for and exploited, reuse can provide effective leverage to a manager when applied to the following areas:

- Architectures
- Specifications

- Requirements
- System design
- Software.

The concept of reuse has existed for many years. The COSMIC Repository[5] started by NASA over a decade ago to make computer programs available to the public, formalized the reuse repository concept. The NASA monthly publication entitled "NASA Tech Briefs" continues to identify and regularly update the reusable components available and new releases (including new technologies) included the NASA COSMIC Repository.

Over the years, reuse has been recognized as providing both leverage and an additional burden and cost factor to program managers; however, true cost savings can be achieved when reuse initiatives are invoked early in the system life-cycle, when designs and architectures are being developed. While the potential savings to be accrued by developing under a reuse paradigm can be significant, it should be noted that supporting standards are virtually nonexistent, and accompanying program management guidebooks on reuse are in their infancy.

3.21.1 DoD Reuse Repositories

In recognition of the dual nature of reuse and in an effort to contain costs, DoD has established and is continuing to establish reuse repositories. The initial efforts focused on identifying software (i.e., code) for inclusion in the repositories. Subsequently, life-cycle data collected over the years and on various projects revealed that greater leverage from reuse could be obtained if reusable components, other than code, could be included in such repositories (e.g., architectural components, design, specifications, requirements). Reusable components fall into three basic categories: 1) use of the reusable component as-is, without any modifications; 2) use of a parameterized reusable component (i.e., can be used within the range of parameterized inputs or outputs); and 3) modification or redesign of a reusable component. In all cases, basic concerns about issues of liability and warranties have surfaced and must be answered before a reusable component is employed in a program. Statistics on the extent of prior usage and previous histories of the reusable component may provide a measure of added confidence when using the particular item. Identification of reuse metrics also provides insight to subsequent use of reusable components and corporate histories (see Section 3.20).

Additionally, the introduction of formal software engineering methods and techniques into the systems engineering arena has provided program managers with additional analytical and reusable capabilities. The introduction of formal languages (i.e., supported by a syntax) and methodologies into systems engineering has provided the capability to develop other system reusable components in a quantifiable and classifiable manner for repository inclusion and subsequent exploitation. Extending classification schema from repository to other engineering

[5] The COSMIC Repository resides at the University of Georgia, 382 East Broad Street, Athens, Georgia 30602, Phone (706) 542-3265.

areas (e.g., hardware, firmware) can provide more extensive repositories. Significant productivity and cost savings across the life-cycle may also result from the timely construction of prototypes (containing design, hardware, and software) that mirror the target system and its requirements very closely.

A current listing of key reuse repositories within the DoD can be found in the *Information Technology Standards Guidance (ITSG)* document, which supports TAFIM Volume 7.

3.22 QUALITY ASSURANCE

Development and execution of a Quality Assurance (QA) program is the responsibility of the program manager. QA program objectives are to: 1) ensure mission and operational effectiveness, user performance, and ownership satisfaction with DoD products; 2) ensure all services and products meet mission and operational needs; 3) ensure essential functional performance and related physical requirements are consistent with needs; 4) ensure contractual requirements are tailored in compliance with DoD direction for specifications and standards; and 5) ensure the other four objectives are cost-effective.

Quality assurance is also the responsibility of all program participants and a requirement of the FAR, which requires the contractor to ensure total contract conformance (product design, manufacture, verification, and delivery). In addition to the contractor, two other independent organizations are involved in QA functions: the Government contracting administration and the program management office. Contract administration or the contracting office is responsible for performing procurement QA, which encompasses accepting the contractor's verification system or quality program, ensuring compliance with all contract requirements, evaluating evidence of product conformance, and performing verification of product conformance before final acceptance. The program office is responsible for ensuring user needs have been translated into enforceable design-to or build-to requirements; participation in design and production readiness reviews; and evaluation of contractor performance in meeting functional and physical uniformity requirements.

Contract provisions for quality include contractor inspection provisions, as on some COTS items and the Standard Inspection Clause, which gives the contractor responsibility for all inspections and tests necessary to ensure contract conformance. The Government may reserve the right to perform any or all inspections and tests before acceptance or to request contractor records for verification. Other higher-level requirements include MIL-I-45208A, *Inspection System Requirement*, used in conjunction with the Standard Inspection Clause, which requires the contractor to establish and maintain a formal, documented inspection system, including vendor control. MIL-Q-9858A, *Quality Program Requirements*, also used in conjunction with the Standard Inspection Clause, obligates the contractor to have a formal quality program. The ISO 9000 series (including ISO Standards 9001 through 9004) describes and clarifies quality concepts and provides guidelines for the selection and use of the other related standards, which identify requirements for a quality management system.

ISO 9001 covers design, development, production, installation, and servicing. The ISO 9002 examines the manufacturer's capabilities in production and installation only, and ISO 9003 focuses on final inspection and testing procedures. ISO 9004 examines each of the quality-system elements in ISO 9000 to help manufacturers set up a quality system; however, this standard is for guidance and should not be contractually imposed.

Quality assurance is also the responsibility of all program participants and a requirement of the FAR, which requires the contractor to ensure total contract conformance (product design, manufacture, verification, and delivery). In addition to the contractor, two other independent organizations are involved in QA functions: the Government contracting administration and the program management office. Contract administration or the contracting office is responsible for performing procurement QA, which encompasses accepting the contractor's verification system or quality program, ensuring compliance with all contract requirements, evaluating evidence of product conformance, and performing verification of product conformance before final acceptance. The program office is responsible for ensuring user needs have been translated into enforceable design-to or build-to requirements; participation in design and production readiness reviews; and evaluation of contractor performance in meeting functional and physical uniformity requirements.

Contract provisions for quality include contractor inspection provisions, as on some COTS items and the Standard Inspection Clause, which gives the contractor responsibility for all inspections and tests necessary to ensure contract conformance. The Government may reserve the right to perform any or all inspections and tests before acceptance or to request contractor records for verification. Other higher-level requirements include MIL-I-45208A, *Inspection System Requirement*, used in conjunction with the Standard Inspection Clause, which requires the contractor to establish and maintain a formal, documented inspection system, including vendor control. MIL-Q-9858A, *Quality Program Requirements*, also used in conjunction with the Standard Inspection Clause, obligates the contractor to have a formal quality program. The ISO 9000 series (including ISO Standards 9001 through 9004) describes and clarifies quality concepts and provides guidelines for the selection and use of the other related standards, which identify requirements for a quality management system.

ISO 9001 covers design, development, production, installation, and servicing. The ISO 9002 examines the manufacturer's capabilities in production and installation only, and ISO 9003 focuses on final inspection and testing procedures. ISO 9004 examines each of the quality-system elements in ISO 9000 to help manufacturers set up a quality system; however, this standard is for guidance and should not be contractually imposed.

This page intentionally left blank.

APPENDIX A

ACRONYMS

A&T	Acquisition and Technology
ADP	Automated Data Processing
AFR	Air Force Regulation
AIS	Automated Information System
AITS	Adopted Information Technology Standards
AMSDL	Acquisition Management Systems and Data Requirements List
ANSI	American National Standards Institute
AR	[1] Adjunct Requirement
	[2] Army Regulation
ASCII	American Standard Code for Information Interchange
ASD	Assistant Secretary of Defense
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computer and Intelligence
CASE	Computer-Assisted Software Engineering
CCB	Configuration Control Board
CDAd	Component Data Administrator
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CDS	Concept Design Sheet
CI	Configuration Item
CIM	Corporate Information Management
CISS	Center for Information System Security
CLIN	Contract Line Item Number
CM	Configuration Management
COTS	Commercial-off-the-Shelf
CPR	Cost Performance Report
C/SCSC	Cost/Schedule Control System Criteria
C/SSR	Cost/Schedule Status Report
DAB	Defense Acquisition Board
DAE	Defense Acquisition Executive
DBMS	Database Management System
DDDS	Defense Data Dictionary System
DEPSECDEF	Deputy Secretary of Defense
DFARS	Defense Federal Acquisition Regulations
DGSA	DoD Goal Security Architecture
DID	Data Item Description

DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DISN	Defense Information System Network
DISSP	Defense Information Systems Security Program
DoD	Department of Defense
DoD DAd	DoD Data Administrator
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
DSSP	Defense Standardization and Specification Program
DT&E	Developmental Test and Evaluation
ECP	Engineering Change Proposal
EDM	Enterprise Data Model
EEO	Equal Employment Opportunity
EI	Enterprise Integration
FAR	Federal Acquisition Regulation
FDAd	Functional Area Data Administrator
FEA	Functional Economic Analyses
FIPS	Federal Information Processing Standard
FIS	Facility Interface Sheet
FMECA	Failure Modes Effects and Criticality Analysis
FPI	Functional Process Improvement
FQR	Functional Qualification Review
HCI	Human Computer Interface
HDBK	Handbook
I-CASE	Integrated Computer-Assisted Manufacturing
ICD	Interface Control Document
ICWG	Interface Control Working Group
IDEF	ICAM Definition Method for Integrated Computer System Manufacturing
IEEE	Institute of Electrical and Electronic Engineers
ILS	Integrated Logistics Support
ILSP	Integrated Logistics Support Plan
IM	Information Management
IPR	In-Process Review
IRDS	Information Resource Dictionary System
ISO	International Organization for Standardization
IT	Information Technology
ITSG	Information Technology Standards Guidance

LAN	Local Area Network
LCC	Life-Cycle Cost
LCM	Life-Cycle Management
LSA	Logistics Support Analysis
LSAR	Logistics Support Analysis Record
MAISRC	Major Automated Information System Review Council
MDA	Milestone Decision Authority
MIL	Military
MNS	Mission Need Statement
NCSC	National Computer Security Center
NDI	Non-Developmental Item
NGCR	Next Generation Computer Resources
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OMB	Office of Management and Budget
OSD	Office of the Secretary of Defense
OSE	Open Systems Environment
OT&E	Operational Test and Evaluation
OTA	Operational Test Agency
PC	Personal Computer
PDR	Preliminary Design Review
PEO	Program Executive Officer
PMO	Program Management Office
PMP	Program Management Plan
PMS	Program Master Schedule
POSIX	Portable Operating System Interface
PRR	Production Readiness Review
PSA	Principal Staff Assistant
QA	Quality Assurance
RAS	Requirements Allocation Sheet
RFP	Request for Proposal
RMP	Risk Management Plan
SBA	Standards-Based Architecture
SBD	Schematic Block Diagram
SDM	System Decision Memorandum
SDP	System Decision Paper
SDR	System Design Review
SECDEF	Secretary of Defense

SECNAVINST	Secretary of the Navy Instruction
SEMP	Systems Engineering Management Plan
SIA	System Interface Agreement
SOW	Statement of Work
SPE	Software Performance Engineering
SRR	System Requirements Review
SSA	Source Selection Authority
SSAC	Source Selection Advisory Council
SSEB	Source Selection Evaluation Board
SSP	Source Selection Plan
SSR	Software Specification Review
STD	Standard
T&E	Test and Evaluation
TAFIM	Technical Architecture Framework for Information Management
TDP	Technical Data Package
TEMP	Test and Evaluation Master Plan
TLS	Timeline Sheet
TPM	Technical Performance Measurement
TRM	Technical Reference Model
TRR	Test Readiness Review
TRS	Test Requirements Sheet
TSR	Trade Study Report
USD(A)	Under Secretary of Defense for Acquisition
WAN	Wide Area Network
WBS	Work Breakdown Structure

APPENDIX B

DEFINITIONS

- To Be Provided -

This page is intentionally left blank.

APPENDIX C

REFERENCES

Note: References appearing in this section represent documents used in preparation of the TAFIM, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

Federal Regulations

Federal Acquisition Regulation (FAR)

OMB Circular A-76, Supplement 1, Cost Comparison Handbook

OMB Circular A-109, Major System Acquisitions

Defense Federal Acquisition Regulation (DFAR)

DoD Directives (DoDD), Instructions (DoDI), and Manuals (in document number order)

DoDD 4105.62	*Selection of Contractual Sources for Major Defense Systems*
DoDD 4120.3	*Defense Standardization and Specification Program*
DoD 4120.3-M	*Defense Standardization Program and Policies, Procedures, and Instructions*
DoD 4245.3	*Design to Cost Manual*
DoDD 4245.7	*Transition from Development to Production*
DoD 4245.7-M	*Transition from Development to Production*
DoDD 5000.1	*Defense Acquisition*
DoD 5000.19-L	*Acquisition Management Systems and Data Requirements List (AMSDL)*
DoDI 5000.2	*Mandatory Procedures for Major Defense Acquisition programs (MDAPS) and Major Automated Information System (MAIS) Acquisition Programs*
DoDI 5000.38	*Production Readiness Reviews*

DoDD 5000.40	*Reliability and Maintainability*
DoDD 5000.43	*Acquisition Streamlining*
DoDD 5000.49	*Defense Acquisition Board*
DoD 5000.52-M	*Career Development Program for Acquisition Personnel Manual*
DoDD 5137.1	*Assistant Secretary of Defense, Command, Control, Communications, and Intelligence*
DoDD 5200.1-R	*Information Security Program Regulation*
DoDD 5200.28	*Security Requirements for Automated Information Systems (AIS)*
DoDD 5200.28-M	*ADP Security Manual*
DoDD 5200.5	*Communications Security*
DoDI 7000.2	*Performance Measurement for Selected Acquisitions*
DoDI 7000.10	*Contract Cost Performance, Funds Status, and Cost/Schedule Status Reports*
DoDD 8000.1	*Defense Information Management (IM) Program*
DoD 8020.1-M	*Interim Management Guidance on Functional Process Improvement* (with Change 1)
DoDD 8120.1	*Life-Cycle Management (LCM) of Automated Information Systems (AISs)*
DoDI 8120.2	*Automated Information System (AIS) Life-Cycle Management (LCM) Process, Review, and Milestone Approval Procedures*
DoD 8120.2-M	*Automated Information System Life-Cycle Management Manual*, Draft
DoDD 8320.1	*DoD Data Administration*
DoD 8320.1-M	*Data Administration Procedures*
DoD 8320.1-M-1	*Data Element Standardization Procedures*
DoD 8320.1-M-X	*DoD Enterprise Data Model Development, Approval, and Maintenance Procedures*

DoD and Military Standards (in document number order)

DoD 5200.28-STD	*Trusted Computer System Evaluation Criteria*
MIL-STD-470	*Maintainability Program Requirements for Systems and Equipment*
MIL-STD 490A	*Specification Practices*
MIL-STD-498	*Software Development and Documentation*
MIL-STD-785	*Reliability Program for System and Equipment Development and Production*
MIL-STD-881	*Work Breakdown Structures for Defense Material Items*
MIL-STD-882	*System Safety Program Requirements*
MIL-STD-973	*Configuration Management*
MIL-STD-1388-1A	*Logistics Support Analysis*
MIL-STD-1388-2A/2B	*DoD Requirements for a Logistics Support Analysis Record*
MIL-STD-1472D	*Human Engineering Design Criteria for Military Systems, Equipment and Facilities*
MIL-STD-46855	*Human Engineering Requirements for Military Systems, Equipment and Facilities*

Military Regulations and Instructions (in document number order)

AFR 70-15	"Proposal Evaluation and Source Selection"
AFR 800-11	"Life-Cycle Costing"
AR 715-6	"Proposal Evaluation and Source Selection"
SECNAVINST 4200.33*S*	"Selection of Contractual Sources for Major Defense Systems"

DoD/Military Handbooks (in document number order)

DoD-HDBK-248	*Guidance for Application and Tailoring of Requirements for Defense Material Acquisitions*
MIL-HDBK-61	*Configuration Management Guide*
MIL-HDBK-71A	*Human Engineering Guidelines for Management Information Systems*
MIL-HDBK-245	*Preparation of Statement of Work (SOW)*

Military Specifications (in document number order)

MIL-I-45208A	*Inspection System Requirements*
MIL-T-31000	*Technical Data Packages, General Specification for Int. Amendment 1 (OSD)*
MIL-Q-9858A	*Quality Program Requirements*

Industry Standards (in document number order)

ANSI/IEEE 1042-1987	*Guide to Software Configuration Management*
ANSI/IEEE 828-1990	*Software Configuration Management Plans*
IEEE 1220	*Standard for System Engineering, Draft Rev 1.0,* Institute of Electrical and Electronic Engineers, April 25, 1994
EIA/IS-649	National Consensus Standard for Configuration Management
ISO 9000/ANSI/ASQC 90	*Quality Standards*
ISO 9001	*Model for Quality Assurance in Design/Development/Production, Installation and Servicing*
ISO 9002	*Model for Quality Assurance in Production and Installation*
ISO 9003	*Model for Quality Assurance in Final Inspection and Test*
ISO 9004	*Quality Management and Quality System Elements — Guidelines*

Publications (alphabetically, by title)

Acquisition and Technology (A&T) Architecture Development Handbook, DISA, Draft, March 31, 1995

Acquisition and Technology (A&T) CIM/EI Program Management Structure, DISA, Working Draft, June 12, 1995

Application Portability Profile (APP), The U.S. Government's Open System Environment Profile Version 3.0 (supersedes NIST SP 500-210), NIST Special Publication 500-XXX, Draft, April 12, 1995

Acquisition How To Guide, DISA, August 1993

Architecture Relationships and Definitions, DISA, Draft, June 20, 1995

Defense Information Infrastructure (DII) Strategic Enterprise Architecture, DISA, Coordination Draft, May 31, 1995

DoD Architectures Review, Draft Technical Report, Volume I (abridged), January 30, 1995

DoD Architectures Review, Draft Technical Report, Volume II (unabridged), January 30, 1995

DoD Corporate Information Management for the 21st Century, a DoD Strategic Plan, Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (C3I), June 1994

DoD Enterprise Integration (EI) Implementing Strategy, DISA Center for Integration and Interoperability, June 1994

DoD Software Performance Engineering (SPE) Project, DISA Center for Standards, Draft, July 1995

DoD Software Reuse Initiative Strategic Plan, DISA, June 1995

GCCS Common Operating Environment Requirements, DISA, Draft, August 15, 1994

Guide on Open System Environment Procurement, Gary E. Fisher, NIST Special Publication 500-220, October 1994

Information Technology Standards Guidance (ITSG), Draft, May 31, 1995

NASA Tech Briefs, NASA Digest Publication, Monthly

Next Generation Computer Resources (NGCR) Acquisition Guide, Space and Navel Warfare Systems Command, SPAWAR 331, NGCR Document No. AST 001 ver. 0.11, Draft, March 30, 1995

Practical Software Measurement, Joint Logistic Commanders, JPCGCRM, Draft Coordination Version, April 12, 1995

Software and Performance Metrics Assessment, DISA, Center for Standards, Draft, August 1995

Software Reuse Implementation Guide, Dept. of the Navy, Naval Information Systems Management Center, Draft, May 1993

Structured Management Process for Architecture Development, DISA, Draft, March 31, 1995

Technical Standards for Command and Control Information Systems (CCISs) and Information Technology, NATO, ATCCIS Working Paper 25, Edition 4, February 25, 1994

Memoranda and White Papers (in reverse chronological order)

"Architecture Terms and Definitions," George Endicott and Anthony Simon, OASD(C3I)/CISA, White Paper, June 30, 1995

"Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement," OASD(C3I), Memorandum (with attachment), October 13, 1993

"Selection of Migration Systems," OASD(C3I), Memorandum, January 15, 1993

"Enhancing Defense Standardization-Specifications and Standards: Cornerstones of Quality," Report to SECDEF by USD(A), November 1988

"Acquisition Streamlining," DepSecDef Memorandum, June 3, 1985

APPENDIX D

TAFIM POLICY MEMORANDA

D.1 This appendix contains the text of the following pertinent policy documents addressing the use of the TAFIM as direction and guidance in the evolution of the DoD Technical Infrastructure.

- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum (with attachment), "Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement," 13 October 1993.
- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum, "Selection of Migration Systems," 12 November 1993.
- Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Memorandum, "Technical Architecture Framework for Information Management (TAFIM)," 30 March 1995.

MEMORANDUM FROM
THE DEPUTY SECRETARY OF DEFENSE

13 October 1993

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
ASSISTANT TO SECRETARIES OF DEFENSE
COMPTROLLER
GENERAL COUNSEL
INSPECTOR GENERAL
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTOR OF ADMINISTRATION AND MANAGEMENT
DIRECTORS OF THE DEFENSE AGENCIES

SUBJECT: Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement

My May 7, 1993, memorandum reiterated the full commitment of the Department of Defense (DoD) to the "...improvements, efficiencies, and productivity that are the essence of CIM." The focus of Corporate Information Management (CIM) on functional process improvement, migration systems, and data standardization has my full support. We need to get on with the job. In order to offset our declining resources, we must accelerate the pace at which we define standard baseline process and data requirements, select and deploy migration systems, implement data standardization, and conduct functional process improvement reviews and assessments (business process re-engineering) within and across all functions of the Department. The acceleration of these actions is key to containing the functional costs of performing the DoD mission within our constrained budget.

The attached guidance requires that addressees expedite selection of standard migration systems and standard data as the basis for process improvement reviews and assessments. The attached guidance expands on direction previously issued by the Comptroller on June 25, 1990, and by the Assistant Secretary of Defense Command, Control, Communications, and Intelligence(ASD(C^3I) on February 11, 1991. The ASD(C^3I) will work with you to ensure that overall functional and Component requirements are met and balanced as we integrate and improve systems, data, and processes across the DoD. Our near-term strategy requires:

- Selection of migration systems within six months, with follow-on DoD-wide transition to the selected systems over a period not to exceed three years.

- Complete data standardization within three years by simplifying data standardization procedures, reverse engineering data requirements in approved and proposed migration systems, and adopting standard data previously established by individual functions and Components for DoD-wide use wherever practical.

The above actions should be implemented immediately, and given appropriate priority in your current and future resource planning and allocation.

Ongoing information management initiatives such as functional process improvement projects, functional and technical integration analysis and planning, and software engineering methods modernization should continue on an expedited basis. However, completion of these current initiatives will not be prerequisites to implementation of the migration system and data standards acceleration strategy. Once standard DoD-wide process, system, and data baselines are established, process improvement studies will be more productive and study results can be more rapidly implemented.

It is understood that the implementation of standard migration systems may result in the loss of automated functionality by selected system users, whereas others may gain functionality. Loss of functionality should not be used as a reason to delay migration system selection and deployment unless there is a documented adverse impact on readiness within the deployment period, or an inability to comply with the law.

The ASD(C^3I) is responsible for supplementing existing procedures with generic evaluation criteria within 30 days to be used in selecting migration systems, and ensuring the objectivity of the selection process.

I request that you personally ensure these actions are accomplished on schedule, and that you report to me on your progress by January 31, 1994.

s/William J. Perry

Attachment

DEPARTMENT OF DEFENSE

STRATEGY FOR ACCELERATION OF MIGRATION SYSTEMS AND DATA STANDARDS

OBJECTIVE

Improve the quality and utility of DoD information while reducing the annual cost of DoD operations.

STRATEGY

Migration Systems

- OSD Principal Staff Assistants, together with their Defense Component counterparts, will, by March 31, 1994, select an information system(s) for each of their respective functional areas of responsibility for designation as the standard, DoD-wide migration system.

- Concurrently, OSD Principal Staff Assistants will develop plans to transition all information technology services throughout the DoD to the selected migration systems, over a period not to exceed three years. Draft plans will be circulated to other Principal Staff Assistants and to Defense Components so that cross-functional and other implementation issues can be identified for consideration by functional and Defense Component members of the DoD corporate Functional Integration Board, chaired by the Deputy Assistant Secretary of Defense (Information Management).

- Funding for development, modernization, or enhancement of legacy systems not selected to be migration systems will be stopped except where approved by the DoD Senior Information Management Official as absolutely essential to support DoD missions or comply with the law.

- The plan for implementing and transitioning services to the selected migration systems should simultaneously forecast a schedule, to the extent practical, for incorporating within the migration systems:

  - Improved functionality and cross-functional integration based on accelerated process improvement reviews and assessments.

  - Interoperability, technical integration, DoD standard data, and integrated databases to provide higher quality and lower cost information technology services for all users.

- Where a requirement is demonstrated to develop a follow-on, new start system to replace the standard migration system in order to meet CIM objectives and the information management policies and principles established in DoD Directive 8000.1, OSD Principal

Staff Assistants will conduct the necessary process improvement studies to develop functional requirements within the next three years.

Data Standardization

- Each DoD Principal Staff Assistant, together with their Defense Component counterparts, will develop and execute a plan in accordance with DoD Directive 8320.1 to standardize the data elements for which they are the custodian within the next three years.
- The ASD(C^3I) will, by January 31, 1994, develop simplified and streamlined processes for data standardization and data administration within the DoD.
- In the interim, the Department will continue to use the existing standard data elements within each function and Defense Component that have been developed under previous procedures. These interim standard data elements are the data standards until replaced by those prepared under DoD Directive 8320.1.

DEFINITIONS

The definitions below are intended to clarify the terms used in the DoD near-term strategy for acceleration of migration systems and data standards. Formal definitions are published in DoD directives or other publications.

Baseline Processes and Data

A baseline is something that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. Baseline processes and data establish how a function operates today (the "as is" environment), and what current functional requirements must be satisfied by the supporting migration system. Process improvement projects assess the "as is" baseline to determine what improvements should be made (to the "to be" environment). Once these improvements have been implemented, they define a new process and data baseline for the next iteration of improvements.

Data Standard (also called standard data)

A data element that has been through a formal analysis (called "data standardization") to reach agreement on its name, meaning, and characteristics, as well as its relationship to other standard data elements. Much like a common language, data standards enable processes and their supporting information systems to be integrated across functions, as well as within them, and improve the quality as well as the productivity of enterprise performance.

Data Standardization

The process of reviewing and documenting the names, meanings, and characteristics of data elements so that all users of the data have a common, shared understanding of it.

Data standardization is a critical part of the DoD Data Administration Program, managed under DoD Directive 8320.1. Data administration is the function that manages the definition and organization of the Department's data.

Function

Appropriate or assigned duties, responsibilities, and tasks that produce products or provide services. In the DoD, a functional area (e.g., personnel) is comprised of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviewing candidates). The functions of the DoD are the responsibility of designated officials who exercise authority over organizations set up to accomplish their assigned functions. The structure and interrelationships among DoD functions and standard data are documented in the DoD Enterprise Model.

Individual functions within the DoD rely on other functions for products and services. In a large, complex enterprise such as the Department of Defense, functions must work together to support the mission of the enterprise; this significantly increases the importance of cross-functional programs, such as data standardization.

Functional Process Improvement (also called business process re-engineering)

Application of a structured methodology to define a function's objectives and a strategy for achieving those objectives; its "as is" and "to be" process and data environments; its current and future mission needs and end user requirements; and a program of incremental and evolutionary improvements to processes, data, and supporting migration systems that are implemented through functional, technical, and economic analysis and decision-making.

Procedures for conducting process improvement reviews and assessments in the DoD are provided in OASD(C^3I) memoranda on Interim Management Guidance on Functional Process Improvement (August 5, 1992, and January 15, 1993).

Integration

Explicit top management initiatives to ensure that interdependent functions or systems operate effectively and efficiently for the overall benefit of the enterprise (i.e., the DoD). This contrasts with coordination among functions or systems, which ensures non-interference, but does not provide integration.

"Integration" implies seamless, transparent operation based on a shared or commonly-derived architecture (functional or technical) and standard data. "Interoperability" implies only the ability of a function or system to exchange information or services with another, separate function or system using translators or interchange rules/standards.

Migration System

An existing automated information system (AIS), or a planned and approved AIS, that has been officially designated as the single AIS to support standard processes for a function. Other AISs, called "legacy systems," that duplicate the support services provided by the migration system are terminated, so that all future AIS development and modernization can be applied to the migration system. A migration system is designated (or selected) by the OSD Principal Staff Assistant(s) and their Defense Component counterparts whose function(s) the system supports, with the coordination of the DoD Senior Information Management Official.

Upon selection and deployment, the migration system becomes the single AIS baseline for:

- Incremental and evolutionary changes that are required to implement functional process improvements, or to execute additional responsibilities assigned to the function that the system supports.

- Technical enhancements that implement standard data and integrated databases, and that migrate the system toward an open systems environment and a standards-based architecture defined by the DoD Technical Architecture Framework for Information Management.

Requirements for selection of migration systems are identified in Chapters 6 and 7 of OASD(C^3I) memoranda on Interim Management Guidance for Functional Process Improvement (August 5, 1992, and January 15, 1993); these procedures should be tailored as appropriate to facilitate expeditious selection. Subsequent development and modernization of migration systems is accomplished in accordance with DoD Directive 8120.1 and DoD Instruction 8120.2.

MEMORANDUM FROM
THE ASSISTANT SECRETARY OF DEFENSE

November 12, 1993

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
DIRECTOR, DEFENSE RESEARCH AND ENGINEERING
ASSISTANT SECRETARIES OF DEFENSE
COMPTROLLER
GENERAL COUNSEL
INSPECTOR GENERAL
DIRECTOR, OPERATIONAL TEST AND EVALUATION
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTOR OF ADMINISTRATION AND MANAGEMENT
DIRECTORS OF THE DEFENSE AGENCIES

SUBJECT: Selection of Migration Systems

This memorandum provides the generic evaluation criteria to be used in selection of migration systems as required by the Deputy Secretary of Defense (DEPSECDEF) memorandum of 13 October 1993, "Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement." The Department of Defense (DoD) must improve the quality and effectiveness of information support for our fighting forces, reduce the cost of duplicative processes, eliminate nonessential legacy systems in all functional areas, and minimize the cost and difficulty of information systems technical integration. Information systems are comprised of applications, data and infrastructure. Expedited selection of migration systems has been established by the Deputy Secretary of Defense as a matter of urgency throughout the DoD. Selection shall be based on these four factors:

- Functional: To be selected as a migration system, the information system will have to be based on defined work processes and will have to be based on the degree to which the system meets the information needs of users within and across functional areas. A decision should be generally supported by the functional user community within the DoD Components, including the Chairman of the Joint Chiefs of Staff (CJCS) representing the unified combatant commands.

- Technical: The system can evolve (migrate) to be supported by the integrated, standards-based architecture prescribed for the future Defense Information Infrastructure (DII).

- Programmatic: A functional economic analysis that documents a reasonable range of alternatives that meet both functional and technical objectives is required. The alternatives must be within programmatic constraints (resources, schedules, and acquisition strategy), and justify adopting the migration system to the Department. Given the compressed time frames, the PSAs may elect to base their migration decision on an abbreviated functional economic analysis. Acquisition strategy planning factors will be considered in accordance with Acting ASD(C^3I) memorandum of February 4, 1993, "Acquisition Strategy Planning for CIM Migration Systems."

- Data: The ability to transition to data standards is a fundamental requirement for an information system in order for it to be selected as a migration system. Applications should lend themselves to data sharing within their design. Migration plans must include transition to DoD standard data and shared data concepts.

Migration systems selection procedures and factors are discussed in our Interim Management Guidance on Functional Process Improvement (August 5, 1992, and January 15, 1993). Except where exempted under DoD Directive 8120.1, Section B, the selection procedures apply to all AISs in the Department. This includes all C^3I systems except those specifically and individually exempted by me in accordance with my DoD Senior Information Management (IM) authority under DoD Directives 5137.1 and 8000.1. All information technology services shall be transition to the selected migration systems over a period not to exceed three years, and the legacy systems providing these services shall be terminated. Any funding for development, modernization, or enhancement of these legacy systems requires the approval of the DoD Senior IM Official, in accordance with the DEPSECDEF's memorandum of October 13, 1993. Life-cycle management reviews of migration systems shall also address these candidate legacy systems and data until their termination.

Migration system selection shall be made by the Office of the Secretary of Defense (OSD) Principal Staff Assistant(s) (PSAs), or CJCS, having functional responsibility for the missions and functions supported by the system, with the participation of affected DoD Components. The choice of functional criteria guidance in the selection of migration systems is the responsibility of the PSAs/CJCS. As the DoD Senior IM Official, I shall approve the proposed selection, based on my review of the selecting official's evaluation of technical, programmatic, and data factors. Because technical factors are critical to successful implementation of the DII, I shall have additional studies conducted where appropriate, and I shall withhold my approval where significant issues remain unresolved. Disagreements shall be resolved in accordance with DoD Directive 8000.1, Section E.1.d.

Attached to this memorandum are key technical considerations that must be addressed in the selection process. Assistance in your selection of migration systems and in preparation of the appropriate documentation is available through the Defense Information Systems Agency Center for Integration and Interoperability. If you would like this assistance, please contact Dr. Michael Mestrovich at (703) 756-4740.

s/Emmett Paige, Jr.

Attachment

KEY TECHNICAL FACTORS TO BE CONSIDERED IN THE SELECTION OF MIGRATION SYSTEMS

Technical Factors

Extent to which the candidate legacy automated information system (including Command, Control, Communications and Intelligence (C^3I) systems) currently conforms to, or can evolve (migrate) to conformance with, the open systems environment and standards-based architecture defined by the DoD Technical Architecture Framework for Information Management (TAFIM)[1].

Difficulty, cost, and time line for migrating the system (including its applications, data, and supporting infrastructure) as expeditiously as possible from its current technical environment to conformance with:

- The TAFIM
- DoD standard data, based on the DoD Data Model. The DoD Data Model is a principal component of the DoD Enterprise Model
- Shared use of applications, databases, and the computing and communications infrastructure with other designated migration systems
- Cost effective, timely, secure, and highly reliable support to all functional users from consolidated data processing facilities

Timeliness, completeness, and availability of life-cycle management and supporting documentation, particularly including data and application software documentation

Difficulty, cost, and time line for application of:

- DoD information technology utility services
- Commercial-off-the-shelf (COTS) software, and portable, re-usable software modules
- Ada and computer-aided software engineering (CASE) tools and methods

Current and future interface, interoperability, and integration requirements with other systems and databases within and across all DoD functional activities and functional areas.

Application of Technical Factors

[1] Office of the Assistant Secretary of Defense (C^3I) Memorandum, "Interim Management Guidance on the Technical Architecture Framework for Information Management (TAFIM)," January 15, 1993.

Application of these technical factors results in giving preference to systems that:

- Have been developed using Ada and other "state of the industry" software engineering best practices, are well documented, and are under good configuration control.
- Use current COTS information technology software and hardware, such as data dictionaries and data base management systems, optical disk technology, etc.
- On the whole, are more compliant rather than less compliant with the technical factors listed above, and apply those factors consistently across all systems supporting the functional area.

Assessment and Plans

The selection of a candidate migration AIS must be founded on its functional and technical adequacy. Migration assessment includes a technical analysis of migration candidate systems to ensure legacy applications will meet the information requirements of the functional user and that has the ability to accommodate subsequent functional and technical improvement activities.

A migration plan consisting of functional, technical and data concerns, with programmatic considerations is the start of the process for selecting migration systems. The DoD "Tree" diagrams, a quarterly publication from DISA/Center for Integration and Interoperability (CFII), displays each functional area's decisions for integrating. These "Tree" diagrams will be completed by all functional areas with target dates to depict the Enterprise Integration. The diagrams present an important migration picture but stop short of the migration planning that is necessary for implementation. The DISA/CFII is available to help each functional area develop migration plans and assess technical cross-functional integration for the Enterprise.

To validate the technical sufficiency of a candidate migration system, the applications should be evaluated in terms of relevant functional, technical, data handling, and programmatic criteria.

MEMORANDUM FROM THE ASSISTANT SECRETARY OF DEFENSE

March 30, 1995

MEMORANDUM FOR
UNDER SECRETARIES OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY (RD&A)
ASSISTANT SECRETARY OF THE NAVY (RD&A)
ASSISTANT SECRETARY OF THE AIR FORCE (ACQUISITION) (SAF/AQ)
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM), Version 2.0

My memorandum dated June 23, 1994 established the TAFIM as the single framework to promote the integration of Department of Defense (DoD) information systems, expanding the opportunities for interoperability and enhancing our capability to manage information resources across the Department. The latest version of the TAFIM, Version 2.0, is complete and fully coordinated. Version 2.0 consists of seven volumes as shown in the attachment. The TAFIM will continue to guide and enhance the evolution of the Department's information systems technical architectures.

I want to reiterate two important points that I made in my June 1994 memorandum. First, the Department remains committed to a long range goal of an open systems environment where interoperability and cross functional integration of our systems and portability/reuseability of our software are key benefits. Second, the further selection and evaluation of migration systems should take into account this long range goal by striving for conformance to the TAFIM to the extent possible.

Effectively immediately, new DoD information systems development and modernization programs will conform to the TAFIM. Evolutionary changes to migration systems will be governed by conformance to the TAFIM.

The TAFIM is maintained by the Defense Information Systems Agency (DISA) and is available electronically via the DISA On-Line Standards Library. Hardcopy is available through the Defense Technical Information Center. The TAFIM is an evolving set of documents and comments for improving may be provided to DISA at any time. The DISA action officer is Mr. Bobby Zoll, (703) 735-3552. The OSD action officer is Mr. Terry Hagle, (703) 604-1486.

s/Emmett Paige, Jr.

Attachment

APPENDIX E

SYSTEMS ENGINEERING ELEMENTS/ACTIVITIES AND PRODUCTS

E.1 The following table identifies and describes the major elements/activities and products of the Systems Engineering discipline discussed in Volume 5, Section 3.15. In addition to the traditional systems engineering elements, the table includes summaries of those engineering disciplines that are considered engineering specialties influencing and supporting the design, development, and operational support of the system. For C4I and information systems programs, engineering specialties may include logistics engineering, reliability and maintainability engineering, human factors engineering, safety engineering, as well as others not included in the table, which are integrated into the system design and development processes through the systems engineering process. The table also includes the governing standards and other resources for each activity that provide more detailed information and guidance on system engineering requirements and implementation.

Table E-I. Systems Engineering Elements/Activities and Products

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Requirements Analysis **See Section 3.3 for the description of Requirements Analysis.**	- System Level Functional Requirements - Performance Requirements - External Interfaces	DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs)and Major Automated Information System (MAIS) Acquisition Programs.*
Functional Analysis/Allocation Forms the foundation for systems engineering and is the method for analyzing performance requirements and devising them into discrete tasks or activities. Involves identification and decomposition of the primary top-level system functions into subfunctions at ever-increasing levels of detail; supports mission analysis in defining functional areas and architectures, sequences, and interfaces; and is used to develop requirements for equipment, software, personnel, and operational procedures to complete implementation and deployment of the system. Should result in a baseline of functions and functional performance requirements, which must be met to adequately accomplish the operation, support, test, and production requirements of the system.	- System Level (Type A) specification - Functional Flow Block Diagrams - N^2 diagram - Timeline Analysis/ Timeline Sheet (TLS) - Mathematical models and computer simulations, if necessary - Requirements Allocation Sheet (RAS), Test Requirements Sheet (TRS), Facility Interface Sheet (FIS), etc. - Logistics Support Analysis Record (LSAR)	MIL-STD 490A, *Specification Practices*; MIL-STD-1388-1A, *Logistics Support Analysis*; MIL-STD-1388-2A/2B, *DoD Requirements for Logistics Support Analysis Record.* DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Design Synthesis and Verification (Conceptual Design) Synthesis is "the performance, configuration, and arrangement of a chosen system and its elements and the technique for their test, support, and operation, all of which to be portrayed in a suitable form such as a set of schematic block diagrams, physical and mathematical models, computer simulations, layouts, detailed drawings, and similar engineering graphics. These portrayals typically illustrate intra- and inter-system and item interfaces, permit traceability between elements at various levels of system detail, and provide the means for complete and comprehensive change control. They are also the basic source of data for developing, updating, and completing the system and configuration items, and for critical item specifications; interface control documentation; consolidated facility requirements; procedural handbooks, and similar forms of instructional data; task loading; operational computer programs; specification trees; and dependent elements of work breakdown structures". Additionally, through synthesis, architectures are transformed from functional to physical; alternative systems concepts, configuration items, and system elements are defined; physical interfaces (internal and external) are defined and refined; and preferred product and process solutions are selected. The results of various technical and design studies as well as requirements delineated from the functional analysis effort are considered in the process, which should take into account the latest technology in the areas of design, producibility, and supportability. Synthesis requires input from all technology and engineering specialty areas that have a bearing on the system or design concept.	- Concept Design Sheet (CDS) - Schematic Block Diagrams (SBD) - Physical or mathematical models - Drawings, specifications, and other technical and supporting documentation.	DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Evaluation and Decision (Trade Studies) This involves continual evaluation and decisions made throughout the design and development activity. Most attractive concepts are selected, evaluated, and optimized. Also, systems engineering identifies and documents the trade-off and supporting rationale and considers all possible solutions within the framework of requirements. (See also Section 3-11 and the Trade Studies/Trade-Off Analyses element, below, in this table.)	- Trade Study Report (TSR)	
Description of System Elements Once an acceptable solution or concept has been selected, interacting system elements are defined, which fall into five categories: 1) equipment/hardware, 2) software, 3) facilities, 4) personnel, and 5) procedural data. Performance, design, and test requirements for equipment end items, critical components, and computer software programs are established and described. Environmental requirements and interface design requirements imposed on facilities by the functional and design characteristics of equipment end items are identified and documented.	- Design Sheets - Facility Interface Sheets	DoD 4245.7-M, *Transition from Development to Production.*
Technical Performance Measurement/Performance Metrics (System Analysis and Control) Defined as the product design assessment that estimates, through engineering analysis and tests, the values of essential performance parameters of the current design of WBS product items. Used to forecast values to be achieved through the planned technical program effort; measure differences between the achieved values and those allocated to the product element by the systems engineering process; and determine the impact of these differences on system effectiveness. Purpose is to	- Contractor Technical Performance Measurement Report	*Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*; DI-S-3619, Technical Performance Measurement Report.

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
provide visibility of actual versus planned performance; provide early detection or prediction of problems that require management attention; and support assessment of the program impact of proposed change alternatives. Alerts program management to potential performance deficiencies before irrevocable cost or schedule impact occurs. Where risk management program is in place, provides data for technical risk planning and assessment. Can begin when configuration item requirements allocation is substantially complete (when draft Type B specifications are available, normally in the demonstration and validation phase). - Also, See Section 3.20, Metrics.		
Interface Management (System Analysis and Control) The documentation, management, and control of functional and performance interface requirements identified during functional analysis. Manages the interfaces within the system and between the system and the outside world; manages requirements as specified in interface control documents; systems engineering chairs Interface Control Working Group (ICWG). (See also Section 3.17)	- Interface Control Documents (ICD)	DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*; MIL-STD-973, *Configuration Management*; *NGCR Acquisition Guide (Draft).*
System Integration The assurance, by systems engineering management, that all diverse elements of a system are compatible and ready when needed. Accomplished through proper planning and coordination through the development process. Basic plan for managing their effort is the Systems Engineering Management Plan (SEMP), prepared in three parts, by the contractor: Part I, "Technical Program Planning and Control", identifies organizational	- Contractor Systems Engineering Management Plan (SEMP)	

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
responsibilities and authority for systems engineering management, including control of subcontracted engineering, verification, configuration management, document management, and plans and schedules for design and technical program reviews; Part II, "Systems Engineering Process", describes the process used in defining and allocating requirements and their documentation; Part III, Engineering Specialty Integration" defines how engineering specialties of reliability, maintainability, human factors engineering, safety, logistics support, and other areas are integrated into the mainstream design effort. SEMP provides the basis for all contractor system engineering efforts, should be program-specific, and should identify the organizational configuration, functions, and responsibilities, management techniques, analyses, trade studies, simulations, Technical Performance Measurement (TPM) parameters, and schedules that will be investigated and employed on the program.		

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Risk Management (System Analysis and Control) Organized means of identifying and measuring risk (risk assessment) and developing, selecting, and managing options (risk analysis) for resolving or handling identified risks. Risk management strategy is established early in the program, and risk is continually addressed throughout the system life-cycle. Risk planning involves articulating program risk issues, identifying risk management strategy and techniques, defining project roles and responsibilities for risk management, developing risk identification, reporting, and tracking procedures. Risk identification involves soliciting risk insight from project personnel, performing risk identification as part of standing review boards, and employing experience from similar projects to identify potential risk. Risk analysis includes characterizing the types and magnitude of risks corresponding to the affected program baseline (technical, cost, schedule risk) and determining and evaluating the probability and impact of risk occurrence possibly through modeling techniques. Some aspects of risk handling include developing a risk avoidance strategy, such as selecting lower-risk technical approaches, choosing to control risk through management attention, transferring risk to another organization, performing research to understand risk sensitivities, and accepting risk as unavoidable. Once identified, risks are monitored and reevaluated until eliminated. Other techniques such as the WBS, TPM, CM, and trade-off analysis may also be considered risk management techniques used for risk assessment and management.	- Risk Management Templates - Contractor and Government Risk Management Plans (RMP) - Contractor Risk Sensitivity Analysis - Contractor Risk Handling Plans - Contractor Risk Reduction Reports - Schedule Network Models - Life-Cycle Cost Model	DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*; DoD 4245.7-2-M, *Transition from Development to Production.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Trade Studies/Trade-Off Analysis (System Analysis and Control) Formal decision analysis method used to solve any complex problem where there is more than one selection criterion and to provide documented decision rationale. Necessary for establishing system configurations and for accomplishing detailed design of individual components. Applicable to budgeting, source selection, test planning, logistics development, production control, and design synthesis. (See also Section 3.11 and the Evaluation and Decision [Trade Studies] activity, above, in this table.)	- Trade-Off Analysis - Utility Curves - Weighted Summary Tables - Trade Study Reports (TSR)	DODI 5000.2, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*; DoD 4245.7-2-M, *Transition from Development to Production*; *NGCR Acquisition Guide (Draft)*.
Reliability Engineering Application of analytical methods and historical statistical data to determine equipment/system performance. Functional models of system performance are derived in accordance with the design, and a mathematical model with outputs of inherent failure distributions and failure rates. By analyzing the design and applying historical data, an estimate of the probability of successful performance (or failure) can be calculated for the system and for each segment, subsystem, assembly, and such. Reliability analysis identifies the strengths and weaknesses of the design, so that improvements can be made to the best advantage. Reliability estimates based on inherent (generic) failure rates are useful for planning purposes, for comparing alternatives, and for assessing proposed changes. Integration of this specialty is important during concept studies, trade-off analysis, design, and development.	- Failure Modes, Effects and Criticality Analysis (FMECA) - Sneak Circuit Analysis - Electronic Parts/Circuits Tolerance Analysis - Reliability Critical Items List - Effects of Functional Testing, Storage, Handling, Packaging, Transportation, and Maintenance - Environmental Stress Screening Report	MIL-STD-785, *Reliability Program for System and Equipment Development and Production.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Maintainability Engineering Addresses the maintenance concept/policy as it is reflected in design provisions for fault prevention, detection, isolation and correction, and the implementation requirements in terms of skills, test equipment, time-to-repair/replace/restore, and maintenance cost over the life-cycle of the system or product. Maintenance concepts are based on operability considerations and on operations phase support concepts. Maintenance provisions are an important design factor in determining system availability and life-cycle cost. Maintainability program plan is normally submitted as part of the bidders' response to the RFP.	- Maintainability Program Plan	MIL-STD-470, *Maintainability Program Requirements for Systems and Equipment.*
Human Systems Integration Addresses people-equipment interfaces. Applies principles of human capability to reach, lift, see, communicate, comprehend, and act to the functions and circumstances required; allocates system functions to personnel, equipment, software, or facilities; identifies level of involvement and criticality of personnel tasks; and performs task analysis and timeline studies to determine if human capabilities will be exceeded. Specialists work with design, system safety, maintainability, testing, training, etc., personnel.	- Human Factors Planing documents and reports - Models and Mock-Ups	MIL-STD-46855, *Human Engineering Requirements for Military Systems, Equipment and Facilities*; MIL-STD-1472, *Human Engineering Design Criteria for Military Systems, Equipment and Facilities*; TAFIM Volume 8, *DoD Human Computer Interface (HCI) Style Guide.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
Specification Development Plays an integral role in the product development process and is the basic critical output of the systems engineering process. The system functional specification (Type A) and expanded lower-level specifications support a proposed technical solution to an approved operational requirement. Specifications applicable to C4I and information systems programs include the following types: **System/Segment (Type A)** states the technical and mission performance requirements for a system as an entity, allocates requirements to functional areas, documents design constraints, and defines interfaces between or among the functional areas. Based on parameters developed during the concept exploration and definition phase. **Development Specifications (Type B, Part I, Design-To)** state requirements for the design and engineering development of a product. Are applicable to an item below the system level and states performance and interface characteristics, and other technical detail sufficient to permit design, engineering for service use, and evaluation. Prepared typically late in the demonstration and validation phase. **Product Specifications (Type C)** are applicable to any level below the system level, and may be oriented toward procurement of a product through specification of primary functional (performance) requirements or primary production (detailed design) requirements. Contain complete performance requirements for intended use, interface and interchangeability characteristics (form, fit, function), detailed description of the product, performance requirements, and corresponding tests and inspections. Prepared in the later part of the development phase. (See also Section 3.14.3.1.)	- System/Segment (Type A) Specification - Development Specification (Type B) - Product Specification (Type C)	Report to SECDEF by USD(A), "Enhancing Defense Standardization-Specifications and Standards: Cornerstones of Quality", November 1988; MIL-STD-490A, *Specification Practices*; DoD 5000.43, *Acquisition Streamlining*; DoD-HDBK-248, *Guidance for Application and Tailoring of Requirements for Defense Material Acquisitions*; DEPSECDEF Memorandum of June 3, 1985, Acquisition Streamlining; DoDD 4120.3, *Defense Standardization and Specification Program*; DoD 4120.3-M, *Defense Standardization Manual*; DoD 4245.7-M, *Transition from Development to Production.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
System Safety Analysis of the system/program for hazards to personnel and equipment and the action taken to eliminate or control them. Encompasses all personnel and equipment that may be affected by program plans and operations. These include, but are not limited to, manufacturing, testing, packaging, handling, transportation, storage, and personnel and equipment at test and operational sites.	- Operational Hazard Analysis - Accidental Risk Assessment Report (ARAR)	MIL-STD-882, *System Safety Program Requirements*.
Configuration Management (CM) Integral part of the systems engineering management process for system definition and baseline management and control. Role is to: 1) identify the functional and physical characteristics of selected system components designated as configuration items; 2) control changes to those characteristics; 3) record and report change processing and implementation status; and 4) coordinate and support design reviews and configuration audits. Means through which the integrity and continuity of the design, engineering, and cost trade-off decisions made between technical performance, producibility, operability, testability, and supportability are recorded, communicated, and controlled by program and functional managers. At any given time, CM can supply current descriptions of developing and operational hardware and software configuration items and the system itself. Provides traceability to previous item and system baseline configurations and rationale for changes, thus permitting analysis and correction of deficiencies. Initiated as early as concept exploration and definition phase, by inputs from systems engineering, and continues throughout the system life-cycle. Provides for the identification and documentation of COTS/NDI, component compatibility, and	- Government and Contractor CM Plans - Configuration Status Accounting Reports - Functional, Allocated, and Product Baseline Listings - Configuration Audit Plans - Configuration Control Board (CCB) Agenda and Minutes	MIL-STD-973, *Configuration Management*; EIA/IS-649, *National Consensus Standard for Configuration Management*; ANSI/IEEE 1042-1987, *Guide to Software Configuration Management*; ANSI/IEEE 828-1990, *Software Configuration Management Plans*;

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
interface, and ensures that the functional characteristics of the system and system performance remain acceptable and documented. CM of COTS products should be done at the form, fit, function level, at the lowest organizational remove and replace level (i.e., LRU). Replacement products should be equivalent at the form, fit, function level. To ensure CM effectiveness, automated CM tools are required, especially for versioning source code and documentation, and the CM manager should report directly to the program manager.		
Technical Reviews (System Analysis and Control) Essential part of systems engineering process and means by which technical requirements and specifications are validated and configuration baselines are established. Can range from very formal technical reviews by Government and contractor systems engineers to very informal reviews involving few personnel and concerned with product and/or task elements of the WBS. Objective is to determine the technical adequacy of the existing design to meet known technical requirements. Reviews become more detailed and definitive as system moves through its life-cycle. The requirements and scheduling of formal reviews is normally included in the SOW of the contract and in the SEMP. They may include: System Requirements Review (SRR), System Design Review (SDR), Preliminary Design Review (PDR), Software Specification Review (SSR), Critical Design Review (CDR), Test Readiness Review (TRR), Functional Qualification Review (FQR), and Production Readiness Review (PRR).	- Technical Review Agenda and Minutes (Contractor) - Contractor's Technical Review Data Package (Contractor	MIL-STD-973, *Configuration Management*; DoDI 5000.38, *Production Readiness Reviews.*

Systems Engineering Elements/Activities	Outputs/Products	Governing Standards/Guidance
The requirements and need for review is controlled by DODI 5000.2, Part 4, "Program Design" , and MIL-STD-973, which should be tailored to factors such as program complexity, level of inherent technical risk, and number of participating contractors.		
Test and Evaluation (T&E) See Section 3.18 for the description of T&E.	See Section 3.18.	DoDD 5000. 1 *Defense Acquisition*; *NGCR Acquisition Guide (Draft).*
Integrated Logistics Support (ILS) See Section 3.19 for the description of ILS.	See Section 3.19	DoDD 5000.1 *Defense Acquisition*;
Producibility N/A - Engineering function directed toward achieving a design compatible with the realities of available manufacturing processes and not considered applicable to C4I and information systems.	N/A	N/A
Life-Cycle Cost Analysis Structured study of life-cycle cost (LCC) estimates and elements to identify life-cycle cost drivers, total cost to the Government, cost risk items, and cost-effective changes. It is a systems engineering tool with application to all elements of the system. Computer modeling is often used to identify and analyze cost drivers, which are areas where resources can best be applied to achieve the greatest benefit in reduced cost. Modeling for LCC is also useful in cost-benefit and cost-effectiveness studies, long-range planning, and budgeting, comparison of competing systems, decisions about replacement of aging equipment, control of an ongoing program, and selection among competing contractors.	- Life-Cycle Cost Reports	OMB Circular A-76, Supplement 1, Cost Comparison Handbook; DoD 4245, *Design to Cost*; AFR 800-11, Life-Cycle Costing.

This page intentionally left blank.

APPENDIX F

OSE INFORMATION SERVICES

F.1 The following table contains a listing of DISA services available for obtaining additional OSE guidance and information pertaining to the TAFIM and related OSE requirements.

- To Be Provided -

This page intentionally left blank.

APPENDIX G

PROGRAM MANAGEMENT RESPONSIBILITIES MATRIX

G.1 The following table identifies the program management areas discussed in Volume 5, the documentation to be produced in relation to each area, and the DoD management level(s) responsible for the products identified.

- To Be Provided -

This page intentionally left blank.

APPENDIX H

PROPOSING CHANGES TO TAFIM VOLUMES

H.1 INTRODUCTION

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submitting proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section H.2. Guidance on the use of the format is provided in Section H.3.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, in the format shown in Section H.2, and provide on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail: **TAFIM**

Booz•Allen & Hamilton Inc.

5201 Leesburg Pike, 4th Floor

Falls Church, VA 22041

Fax: **703/671-7937**; indicate "TAFIM" on cover sheet.

H.2 TAFIM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number:

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

H.3 FORMAT GUIDANCE

The format in Section H.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact with any questions regarding the proposed change. The information in the **Point of Contact Identification** Part (**H.2a**) would identify that individual. The information in the **Document Identification** (**H.2b**) is self-evident, except that a volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** (**H.2c, H.2d, or H.2n**).

In the **Proposed Change #** parts of the format, the Section Number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted; providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: "The working baseline will only be available to the TAFIM project staff." The goal is for the submitter to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The H.2c (5), H.2d (5), or H.2n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, submitters should provide citations from professional publications. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to the improvement of the specific TAFIM document may be provided in H.2 c (6), H.2 d (6), or H.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

This page intentionally left blank.

APPENDIX J

INFORMATION SYSTEM ARCHITECTURE RELATIONSHIPS AND DEFINITIONS

J.1 This appendix has been created to include the definitions being developed by DISA/D5 in the *Information System Architecture Relationships and Definitions* draft document. This document is being staffed separately. This coordinated version will be incorporated in this appendix in the Version 3.0 Final.

-To Be Provided-

This page intentionally left blank.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 6: Department of Defense (DoD) Goal Security Architecture

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This volume, Volume 6 of the TAFIM, has been changed from the previous edition to place a greater emphasis on the specific phases of the system engineering process, and how each feeds into the next. A significant attempt has been made to impose a consistent story-line on abstract and generic architecture views and security allocations for all elements. Additionally, some restructuring of the volume was done to make navigation through the document flow more consistently and coherently. Information pertaining to standards appearing in this volume has been updated to reflect current situations.

The next edition of this volume will be updated as necessary to reflect the DoD policies changes and decisions noted above.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.

- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.

- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. The Defense Information Systems Agency (DISA) is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Other working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

CONTENTS

FIGURES

1.0 INTRODUCTION

The Defense Information Systems Security Program (DISSP) was initiated at the request of the Assistant Secretary of Defense (Command, Control, Communications and Intelligence). The Defense Information Systems Agency (DISA) and the National Security Agency (NSA) agreed to cooperate in achieving eight security objectives. These objectives were in the areas of:

- Security policy
- Architecture
- Standards and protocols
- Accreditation procedures
- Technology
- Transition planning
- Organizational improvement
- Products and services availability.

Accordingly, a DISSP Office was established and among its responsibilities was the development of the Department of Defense (DoD) Goal Security Architecture (DGSA). The DISSP has since become a part of the CISS in DISA. The Center for Information System Security (CISS) assists DoD organizations in the transition of existing systems and in the development of new systems in accordance with the DGSA.

Concurrent with the development of the DGSA, efforts were underway within DISA to define information system architectures for the Defense Information System (DIS). These efforts focused on the Technical Architecture Framework for Information Management (TAFIM). The TAFIM is intended to be generic and sufficiently flexible in its definition so that specific systems may be developed or modified to satisfy specific mission goals. The TAFIM is thereby a "goal information system architecture" and has incorporated the DGSA, as Volume 6, as its "goal security architecture."

1.1 PURPOSE

The DGSA was developed in conjunction and harmony with the total requirements for automated services. The protection of information and system assets was a key consideration as part of the total view of objectives, threats, performance, interoperability, extensibility, usability, and cost of implementations. The DGSA does not provide a specification for any particular information system or component. Rather, it specifies security principles and target security capabilities that will guide system security architects in creating specific security architectures that are consistent with the DGSA. While there is no fixed date by which all aspects of the DGSA will be achieved, the concepts of the DGSA can be applied to information systems today. As security technology improves and products incorporate support for DGSA concepts, specific information systems will achieve greater and greater consistency with their individual goals.

After the initial release of the DGSA, activities were undertaken to create a DGSA Transition Plan (CISS, 1995) to define the steps needed to incorporate DGSA concepts into information systems. The Transition Plan is intended for system planners and managers addressing security in information system development or modernization programs. It may also be used by commercial developers, vendors, and those interested in incorporating specific security initiatives or objectives outlined in the Transition Plan into their product developments or security programs. System security engineers and integrators will be able to take advantage of the development of security products and mechanisms that will result from implementation of the Transition Plan. Like the DGSA, the Transition Plan is a living document that will be updated periodically to take into account changes in technology and new application areas.

1.2 SCOPE

The DISSP was instituted to draw together various information system applications, information transport systems, programs, and architectural activities to bring about consistency, efficiency, and interoperability in the security designs for the DIS. Several programs and systems were identified, such as the Defense Message System (DMS), the Defense Information Systems Network (DISN), the Integrated Tactical/Strategic Data Network (ITSDN), and the DoD Multilevel Security (MLS) Program, as well as emerging applications such as electronic commerce, as candidates from which DISSP personnel could gather a complete set of security requirements. These programs cover the bulk of the DIS and are reasonable representatives of DoD information processing needs as well as those of commercial and Federal communities. The DGSA encompasses this diversity of information systems to achieve greater efficiency and interoperability throughout the DIS and other communities.

1.3 ARCHITECTURAL TYPES

Information system architectures range in definition and occur in sequence from abstract views to specific views of what is to be developed. Experience shows that four types are frequently used: abstract, generic, logical, and specific. The TAFIM is considered to be an abstract and generic architecture and the DGSA, as part of the TAFIM, is also abstract and generic.

1.3.1 Abstract Architecture

An abstract architecture begins with knowledge of the requirements and defines corresponding functions to be performed. It defines principles and fundamental concepts that guide the selection and organization of functions. Abstract security architectures cite principles, fundamental concepts, and functions that satisfy the typical security requirements. These concepts and functions are allocated to elements of an abstract definition of the information system architecture.

1.3.2 Generic Architecture

The development of a generic architecture is based upon the abstract architectural decisions. It defines the general types of components and allowable standards to be used, and identifies any necessary guidelines for their application. A generic security architecture proceeds from an initial allocation of security services and functions and begins to define the types of components and security mechanisms that are available to implement the security services with particular strengths. Any limitations in combining components and mechanisms because of incompatibility or security degradation must be cited in the guidelines for application.

1.3.3 Logical Architecture

A logical architecture is a design that meets a hypothetical set of requirements. It serves as a detailed example that illustrates the results of applying a generic architecture to specific circumstances. The only differences between a logical and a specific architecture are that the specific requirements are real, not hypothetical, and since the logical architecture is not intended to be implemented there is no need to perform a cost analysis. In logical security architectures, the logical design is accompanied by an illustration of the security analysis to be performed in specific architectures.

1.3.4 Specific Architecture

The objective of any system architect is to accomplish a level of design specification such that components may be acquired to implement the system. The specific architecture addresses components, interfaces, standards, performance, and cost. Specific security architectures show how all the selected information security components and mechanisms, including doctrine and supporting security management components, combine to meet the security requirements of the specific system under consideration.

1.4 DOCUMENT ORGANIZATION

Section 2 introduces the broad set of requirements to which the DGSA is responsive. The reflection of these requirements in a security policy and their use within a systems engineering process is discussed. In Section 3, an abstract information system architecture is presented, which includes the identification of major components of a generic information system; an abstract information model is discussed; and security responsibilities are allocated to the major architectural components based upon realistic expectations of the protections that can be achieved. Section 3 also presents several key security concepts used throughout the remainder of this document. The major components identified in Section 3 are then considered in detail, specifically end systems and relay systems in Section 4, security management in Section 5, transfer systems in Section 6, and administrative and environmental security measures in Section 7. Section 8 presents a logical architecture example of the application of the DGSA.

This page intentionally left blank.

2.0 SECURITY POLICY, REQUIREMENTS, AND ARCHITECTURES

This section first discusses the relationships between security policy and security requirements and how they are used within a systems engineering process to create a security architecture. Then, the security policy and security requirements upon which the DGSA are based are presented. Finally, some additional factors which influence security architecture choices are discussed.

2.1 SECURITY POLICY AND SECURITY REQUIREMENTS

Organizations often group their activities within one or more *missions* that focus on some subset of the organization's objectives. An *information system* is a collection of information processing and communications components, and the environment in which they operate, used to support the operations of one or more missions. A *security policy* pertains to organizations and their missions and is based upon the threats to mission accomplishment. A security policy (or, in a more general sense, a collection of security polices) documents the *security requirements* to be placed upon resources used by an organization. These security requirements express, for the organization's personnel, the organization's desired protection for its information and other system resources.

A security architecture designed to meet a specific mission's security requirements defines appropriate security services and mechanisms and allocates them to components of the mission's information system architecture. Since the DGSA is intended to address the needs of all DoD organizations, it is a more general statement about the common collection of services and mechanisms any information system might offer and allocates the security services and mechanisms to the generic components of an information system architecture.

Figure 2-1 shows that security policy and security requirements are derived as a result of examining the threats to a mission and are therefore a subset of the mission's requirements. It also indicates the strong relationship among mission, users, information, and policy. The DoD organizations that will employ the DGSA have many different missions. The security policy addressed by the DGSA is a general expression of the security requirements commonly found among the mission requirements of DoD organizations.

Security requirements are established in the same ways, whether for an entire organization or for a specific mission. The information to be managed is identified; the operational requirements for the use of the information are stated; the value of the information is determined; and the potential threats to the information are identified. Then, the security policy for either the entire organization or a specific mission can be stated in terms of the requirements for:

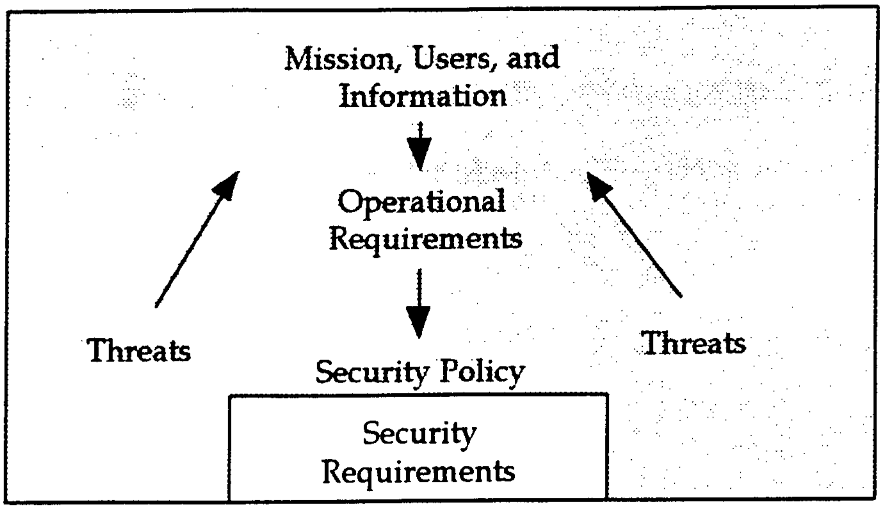


Figure 2-1. Derivation of Security Policy and Security Requirements

- Protection of the information based on the potential threats
- Security services that afford the appropriate protection of the information based upon the value of the information and the threats to it.

2.2 SECURITY ARCHITECTURE DEVELOPMENT

The development of security architectures, whether for entire organizations or for specific missions, are properly part of a larger systems engineering process. The process starts with a mission statement and progresses through a set of well-defined steps that culminate in the deployment and maintenance of information system components that satisfy organizational and mission needs. The first few steps of the process lead to an information system architecture that includes the security architecture. As a result, the security architecture, although separately identified, must be created in conjunction with the information system architecture. Mission-specific information system and security architectures are bounded by architectural decisions made in the higher level organizational architectures.

Figure 2-2 presents the first steps of a security engineering process showing the development of a mission-specific architecture and its relationship to a broader organizational architecture. Starting from a set of DoD requirements for the general DoD mission, a draft DoD Security Policy was created (see Section 2.3) and within the framework provided by the TAFIM, the DGSA was developed. Thus, the DGSA is responsive to the full range of DoD missions.

The development of a mission-specific security architecture begins by applying the DoD security policy to the specific mission requirements in order to develop a mission-specific security policy. The mission-specific security policy includes identification of the appropriate security services

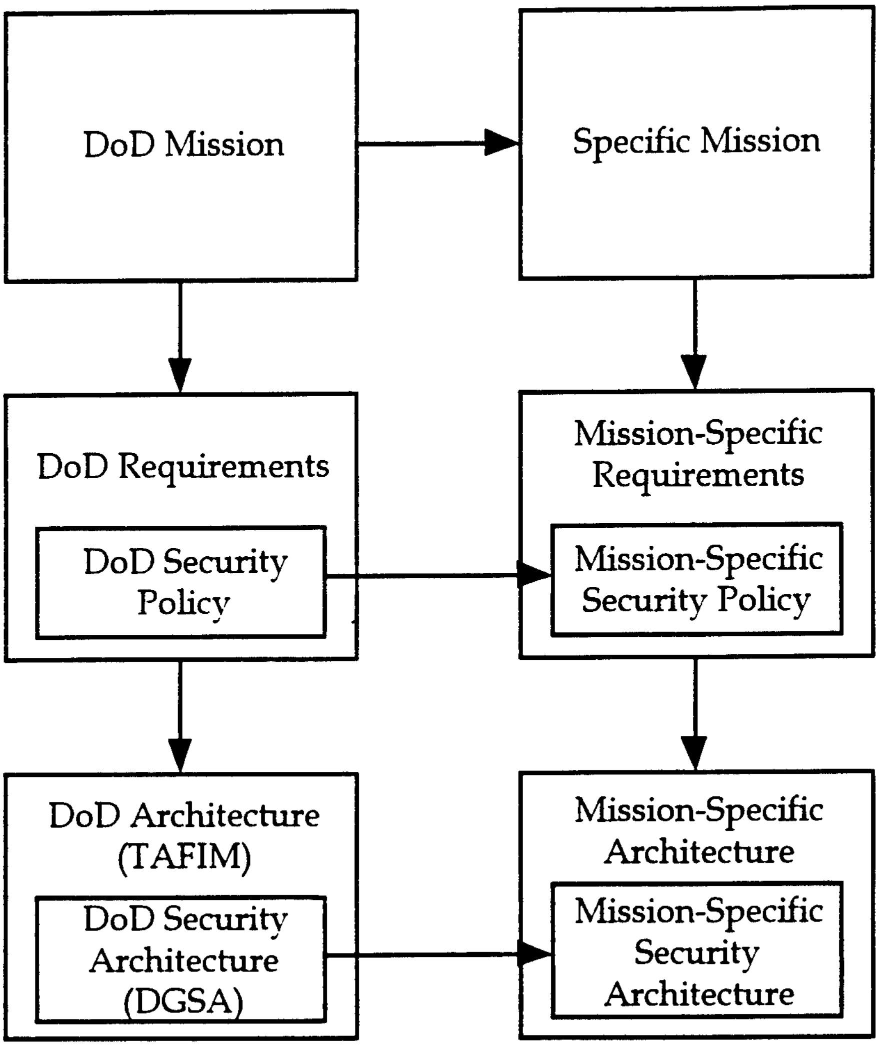

DoD Mission
Specific Mission
DoD Requirements
DoD Security Policy
Mission-Specific Requirements
Mission-Specific Security Policy
DoD Architecture (TAFIM)
DoD Security Architecture (DGSA)
Mission-Specific Architecture
Mission-Specific Security Architecture

Figure 2-2. Mission-Specific Security Architecture Development

an information system needs to satisfy those requirements. The mission-specific information system security architecture is developed using this set of mission requirements and identified security services. The mission-specific architecture is stated as the set of mechanisms appropriate for providing the required protection.

Guidance documents such as the TAFIM, and particularly the DGSA, should be applied to a specific information system architecture to ensure that the necessary security protections are appropriately allocated to specific information system components. Specific security architectures also need to address any applicable policy, public laws, and executive orders. Information system security architects should understand the complete methodology and the way other aspects of the DGSA are taken into account, as demonstrated in the example in Section 8.

2.3 DOD SECURITY POLICY AND SECURITY REQUIREMENTS

1. The DISSP was initiated by appointed panels that studied various aspects of DoD information system security. Their findings and recommendations, including information processing requirements, were collected in the DISSP Action Plan. One of the recommendations resulted in the creation of a draft *DoD Information Systems Security Policy* (NSA, 1993), which is summarized as follows: DoD information systems must support information processing under multiple security policies of any complexity or type, including those for sensitive unclassified information and multiple categories of classified information.

2. DoD information systems must be sufficiently protected to allow distributed information processing (including distributed information system management) among multiple hosts on multiple networks in accordance with open systems architectures.

3. DoD information systems must support information processing among users with different security attributes employing resources with varying degrees of security protection, including users of nonsecure resources if a particular mission so dictates.

4. DoD information systems must be sufficiently protected to allow connectivity via common carrier (public) communications systems.

Notwithstanding the DISSP panels' emphasis on DoD mission requirements, reflection on the activities of other governmental and commercial organizations reveals that these policy statements also are generally applicable to them. Thus, the DGSA is widely applicable outside the DoD.

Analysis of the security policy statements above leads to a set of DGSA security requirements, including multiple information security policy support, open system employment, appropriate security protection, and common security management. These security requirements are presented at a moderate level of abstraction. There is no intention to identify every possible low-level or specific security requirement. It is expected that developers will perform similar, but complete, analyses for specific systems.

2.3.1 Multiple Information Security Policy Support

Some current information systems support simultaneous processing of information at multiple sensitivity levels (e.g., by using multilevel ssecure systems) and others support simultaneous processing of collections of information under the same security policy (e.g., Controlled Mode Workstations). However, no current information systems satisfy the long-held desire by users to operate simultaneously under several different security policies on a single device (e.g., workstation, outboard protocol device). Policy statement 1, above, recognizes that support for multiple security policy operation must become commonplace. The successful implementation of policy statements 1, 3, and 4 largely depends on the ability of information systems to separate information and user activities subject to different security policies. That is, implementations must provide users with confidence that there will be no security policy violations when information systems are shared and the users operate under different security policies.

Security policy enforcement is dependent on the ability of supporting information systems to maintain reliably the identities of users and the identification of information under each security policy. The traditional expression of policy enforcement is that all references by users (or processes representing them) to information must be mediated by a reference monitor. The DGSA adopts and extends the reference monitor concept. (Note that any number of reference monitor implementations may be possible.)

When information processing operations take place in distributed information processing systems, the security policy enforcement for information in transit is commonly supported by mutual authentication, access control, data integrity, data confidentiality, and non-repudiation communications security services. For local (e.g., within a workstation) information processing, a similar set of security services can be applied.

2.3.2 Open Systems Employment

Employment of open systems is a typical operational requirement in many environments. Open system employment is central to providing information security among distributed DoD information systems where simultaneous support of multiple security policies is required. This requirement will lead to increased sharing of processing resources through the operation of a wider variety of applications than seen on current systems. Not only is this requirement directly derived from policy statement 2, but it supports policy statements 3 and 4 as well.

When a user seeks to perform functions in a distributed environment, the user must be able to convey information to another user (or a process) that will become the basis for decisions about what kinds of interaction will be allowed. The DGSA presumes that DoD-approved standard protocols (international or at least national or DoD standards, as opposed to industry proprietary schemes), information, and mechanisms will enable users to determine the capabilities and environment of other users or system processes with which they attempt to communicate. The determination may be made on the basis of information available before any communication is attempted (e.g., from a directory service), or as part of the initial communications service negotiation, or a combination of these approaches. The result of such a determination might be that the only common capability, within the information security policies shared by the users, is to share only non-sensitive information or that no further communication is possible.

Beyond the normal means to begin distributed processing, standards for the representation and exchange of security information are needed. Some of this information is made available as part of the communications exchanges and some is provided through security management-related exchanges. Taken together, this information is used in the provision of various security services.

2.3.3 Appropriate Security Protection

Policy statements 2, 3, and 4 refer to information systems being "sufficiently protected" or supporting users by employing varying degrees of security protection. To protect specific information, an appropriate combination of automated, procedural, and physical methods should be chosen from the complete set employed for a particular information system within a particular environment. The appropriate security protection can only be determined by those persons responsible for the particular information and who are able to assess its value and the threats to it as expressed in the applicable security policy. The corresponding generic DGSA requirement is that specific means must be available to users to invoke security mechanisms appropriate to the task at hand.

What constitutes appropriate security protection, in part, is affected by the security protection provided by the communications system that is used among distributed systems. Policy statement 4 requires that when common carrier communications must be used, the information systems must be prepared to provide all of the appropriate security protection. The only security service that should be assumed from a common carrier communications system is availability.

The requirement for appropriate information systems security protection dictates that security mechanisms must be identified that implement security services at the level of protection required in security policies. Since some security mechanisms may be used to provide (parts of) multiple security services and some security services may be implemented by multiple mechanisms, a determination must be made that the mechanisms are appropriate individually and in combination. Initially, this is a technical activity, but the final determination involves deciding whether shortfalls in the collected security mechanisms can be accepted or whether additional measures must be put in place. This determination must be made by the users of mission information, or as is most common, the designated authority for system operation (accreditor) who represents the users.

2.3.4 Common Security Management

Like the open systems employment requirement, security management appears to be concerned with operational issues, but it actually provides the foundation for many of the security mechanisms that implement the security services chosen to satisfy the other security requirements. Commonality in security management will allow security administrators to control, in a uniform manner, systems that operate under multiple security policies in accordance with policy statements 1 and 2.

The basic elements that must be managed are users, security policies, information, information processing systems that support one or more security policies, and the security functions that support the security mechanisms (automated, physical, personnel, or procedural) used to implement security services. For each of these elements, the managed objects that constitute them must be identified and maintained. For example, users must be known and registered, the security policies must be represented and maintained, and information objects must be identified and maintained. The format for presenting the information in managed objects and operations on them must be standardized. Section 5 presents a detailed discussion of these managed objects and an architecture for security management.

2.4 FACTORS THAT CREATE ADDITIONAL SECURITY REQUIREMENTS

Several factors either directly or indirectly create additional security requirements. This section identifies selected factors that influence security and discusses the security requirements derived from those factors. The selected factors and the derived security requirements from those factors are shown in Figure 2-3. The presentation of the factors is designed to promote a thought or investigative process that should be applied to specific missions.

Operations today must exist in an environment in which major trends tend to be at odds with one another. Technology advancement has provided an opportunity to create an operational vision barely imaginable a few years ago. However, the high cost of transitions and diminishing budgets act against employing the new technologies. Intelligent strategies which may not reduce up-front costs but show valuable long-term benefits and reductions in costs will win favor. These strategies must support the long-term operational objectives of enterprises. Such strategies include portability of applications and other software, continuous upgrades of hardware and software, ensuring scalability of applications and communications resources, reuse of software components, and reuse of certification and accreditation results. Each strategy has the post-transition value of providing low-cost growth paths, if supported properly, and each strategy has an effect on security. Ease of recertification of systems and products after change may be the most important of the strategies in its long-term payoff.

2.4.1 Use of Off-the-Shelf Equipment

Economics have always been a driver in decisions to employ security solutions for information systems. Implementation of automated security measures has raised system costs while providing questionable returns on investments. One of the reasons that costs of security measures have remained high compared to their value is that security measures have been implemented in specialized, often retrofitted, components. Particularly in the face of current budgetary constraints, it is highly desirable that security features become standard elements of commercial-off-the-shelf (COTS) or government-off-the-shelf (GOTS) equipment so that security has minimal impact on price. For this change to occur, vendors must be persuaded to create products with security features that are integral parts of those products. Vendors will need to be convinced that a broad market for such products exists. Evaluation, and certification and accreditation (C&A) must become streamlined and conclusive processes so that vendors can be assured of reasonable returns on investments. Creation of a viable security product market will depend on the use of standards for commercial, international, and DoD use. Availability of COTS and GOTS products with integral security features will affect the ability to satisfy mission security requirements.

SELECTED FACTORS	DERIVED SECURITY REQUIREMENTS
Economics	Security features are standard elements of COTS and GOTS equipment
	Security product standards for commercial, international, and DoD use
	Evaluation and certification and accreditation (C&A) are streamlined and conclusive processes
Information Centralization, Access and Interoperability	Coexistence of varying sensitivities of information on the same information system
	Proper separation, authentication, labeling, and access control
Total Access to All Necessary Information	Improved authentication
	Improved availability
	General secure display implementations
Information Separation While Systems and Information are Shared Among Enclaves	Mechanisms that allow shared systems and information among enclaves, while ensuring appropriate separation of users and information
Increased Connectivity Without Increased Cost	Security mechanisms adequate to protect information from hostile entities on a network
	Standards for security protocols, authentication information, key management and distribution, security management information, voice communications, and methods to evaluate protection
Increased Access to Information and Resources	Interoperability of communications and security services
	Establishment and separation of enclaves
	Interpretation and exchange of security information in standard forms
	Management of security information
Transparency in Distributed Processing	Unitary logon and authentication
Consistent and Uniform C&A Applicable Across DoD Systems and Products	Uniform C&A procedures
	C&A results usable by evaluators and accreditors
	Metrics for effectiveness of security mechanisms
	Metrics for the interaction of a collection of security mechanisms

Figure 2-3. Selected Factors and Security Requirements

2.4.2 Objectives of Enterprise Initiatives

DoD-wide enterprise initiatives, such as the Center for Information Management (CIM) and Command, Control, Communications, Computers, and Intelligence (C4I) for the Warrior (C4IFTW), impose operational objectives that impact security. The CIM promotes information centralization, information access, and interoperability. All three of these operational objectives eliminate consideration of isolated or stand-alone implementations as a means of providing security. The derived security requirements from these objectives are the need to consider both the coexistence of varying sensitivities of information on the same information system and the provision of proper separation, authentication, labeling, and access control. C4IFTW is designed to provide the war-fighting soldier with access to any information needed to do the job, regardless of sensitivity, media, or branch of Service. Such operational objectives provide security challenges and considerations. System interfaces for war-fighting equipment are not equivalent to those for non-war-fighting equipment; thus new authentication issues are raised. Access to the information in a pull-from (information-on-demand) mode emphasizes both interoperability and availability requirements. The integration of voice, imagery, and data requires data correlation and a general secure display (windows) implementation.

While not all of the operational objectives discussed here necessarily pertain to every mission, the implications of the CIM, C4IFTW, and other relevant enterprise initiatives should be considered for their effects on specific missions.

The requirements of specific missions will, in turn, also impose requirements due to specific mission objectives. For example, most missions will require the creation of several groups or enclaves joined together to achieve some specific purpose. It is also likely that the individuals involved will be members of more than one of these enclaves and will need to operate in two or more enclaves simultaneously. Organizations can no longer afford to build separate systems to support each enclave, nor is it effective to require the user to change interface components (such as a workstation) every time the need arises to operate in a different enclave. To achieve the objective of supporting these missions, systems must ensure the separation of information while providing system and information sharing among enclaves. The derived security requirement from this mission-specific objective is to establish criteria for mechanisms that allow multiple enclaves to share systems and information while guaranteeing the separation of information and users as necessary.

2.4.3 Increased Connectivity and Access to Information and Resources

A common and significant operational objective is to take advantage of computer and communications technology to accomplish the mission at hand. This objective can be partially achieved by increasing the potential for connectivity, making additional resources available. Other operational objectives demand that such increased connectivity cannot increase cost significantly. One approach to increased connectivity is to employ commercially available, common carrier networks. However, this approach introduces significant potential risks. There is always the possibility that a hostile entity, with access to the network, will use any means affordable to mount attacks on information systems using the network. A derived security requirement of the operational objective then, is that the security mechanisms chosen to protect information must be adequate to deter such a hostile entity.

Increased connectivity and use of common carrier systems present a perfect environment for DoD-wide interoperability. The connectivity to common carriers will dictate lower layer standard protocols (International Organization for Standardization (ISO) Open Systems Interconnection (OSI) Reference Model (RM), ISO 7498-1 (ISO, 1994a) Layer 3 and below), while the DoD missions will have to address upper layer standards (ISO Layer 4 and above) for interoperability between local environments. This standardization will include authentication information, security protocols, key management and distribution, and security management information. Similar standards for voice communications will also be needed. Additionally, the potential threat of a hostile entity will require standard methods of evaluating the protections afforded to information and other resources to ensure that remote user environments are providing equivalent protection.

As noted in Section 2.4.1, security considerations cause enclaves to arise based on mission criteria that require separation of users and information. Operational objectives, on the other hand, create the need to traverse enclave boundaries. That is, they create a need to provide users with access to any information and resources needed to complete a task. The objective includes operational concepts such as information pull, distributed processing, and information sharing. For example, pull-from may mean information will come from another enclave. Some missions will require support by non-DoD personnel and resources. This requires interoperability of communications and security services. In dealing with access to and the sharing of information and resources, the following derived security requirements must be addressed: establishment and separation of enclaves, interpretation and exchange of security information in standard forms, and management of the security information.

Transparency in distributed processing (i.e., users behaving as if all resources are locally available) is another often stated objective. Users wish to be able to be authenticated once to the local system and then transparently interact with the other systems to access resources. The derived security requirement from this objective is that information systems must have adequate local authentication schemes and security management mechanisms that free the user from the burdens of procedures such as multiple logons.

2.4.4 Achieving Uniform Accreditation

Certification is the process of determining the effectiveness of all security mechanisms. *Accreditation* is the process by which an organization (or an individual on behalf of the organization) accepts or rejects operational responsibility for an information system's performance, including security, in supporting their operations.

Certification and accreditation are complementary procedures that need to be consistent, uniform, and applicable across DoD systems and products. Certification procedures have lacked uniformity and a clear path to completion. In many cases, accreditation procedures are subjective and ad hoc. These deficiencies have caused tremendous frustration on the part of both users and developers of systems. The results of the C&A procedures applied to particular products and systems should be immediately usable by evaluators and accreditors of products and systems that have common elements. The challenge is to develop a set of uniform procedures that establish time limits on the procedure, reduce the time to achieve product and system acceptance, and that will eliminate disparities in the C&A processes. Uniform procedures will ensure consistent and interoperable security support for an organization throughout a distributed environment.

The DGSA concepts presented in Section 3.3 provide a basis for achieving uniform accreditation. Structures and tools for information management are defined that lead to a better understanding of how information is protected, thus making C&A a more tractable endeavor.

This page intentionally left blank.

3.0 SECURITY VIEWS AND CONCEPTS

This section describes abstract and generic security views of information system architectures (Section 3.1). Security service allocations are made to the architectural components identified in these security views (Section 3.2). To accomplish these security service allocations, several concepts are presented that support the DGSA (Section 3.3). These security concepts are used throughout the remainder of the DGSA.

3.1 INFORMATION SYSTEM ARCHITECTURE SECURITY VIEWS

A typical abstract architectural view of the DIS (and many other distributed information systems) divides the information system resources into user elements and network elements (e.g., local area, wide area). This division is a useful starting point for establishing architectural views to which security services can be allocated.

3.1.1 Abstract Information System Architecture Security View

For security purposes, the most useful abstract view of the DIS groups information system resources into *local subscriber environments* (LSEs) that are connected to one another by *communications networks* (CNs). Figure 3-1 illustrates this first security view of distributed information systems.

The LSEs include all devices and communications systems under user (organization) control. The CN provides communications capabilities that allow LSEs to share information. This abstract view is useful for making certain basic security service allocation decisions, but slightly more architectural detail is necessary to make further such allocations.

3.1.2 Generic Information System Architecture Security View

The generic information system architecture security view first refines the abstract LSE and CN into several elements and then defines four generic security architecture components based on the LSE elements and the CN. These architectural components become the focus of the succeeding four sections of the DGSA (Sections 4-7).

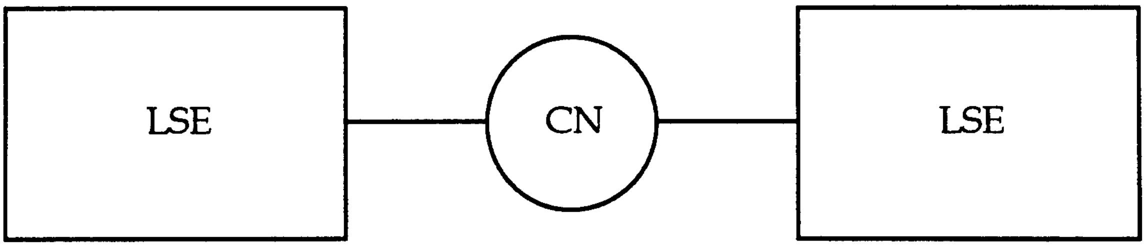


Figure 3-1. Abstract Information System Architecture Security View

3.1.2.1 LSE and CN Descriptions

Included in the LSE are three generic functional elements:

- End systems (ESs) (e.g., workstations, servers, telephones, radios, mainframes)
- Relay systems (RSs) (e.g., multiplexers, routers, switches, cellular nodes, message transfer agents)
- Local communications systems (LCSs) (e.g., rings, buses, wire lines).

The principal distinction between end systems and relay systems (as described in ISO 7498-1) is that end systems support users with direct human interfaces and personal or general applications, while relay systems are only indirectly accessible by users and the functionality is limited to information transfer relay functions. Some relay system functions may be performed in many communications protocol layers (see Section 6). LCSs serve to connect ESs and RSs within an LSE. LCSs may consist of a variety of components, but generally the DGSA is not concerned with specific technologies. Where necessary, the abstract CN can be refined to generic elements such as packet switches, routers, and transmission elements. Generally, the DGSA is independent of particular switching element and transmission technologies, so it is usually adequate to refer to a CN as both an abstract and a generic element.

An LSE may contain a single end system such as a workstation, a single relay system such as a router, or combinations of end systems and relay systems connected through LCSs. All physical elements of the information system architecture are either part of an LSE or are CNs. This security view does not imply that LSEs are only connected to one CN or that they are connected only in pairs.

3.1.2.2 Generic Security Architecture Components

From a security perspective, it is not enough to consider only the physical information system elements. It is necessary to take into account the environment in which the elements are employed and the means through which they are managed. The resulting generic security architecture view includes four components to which security service allocations will be made:

- ESs and RSs - information processing elements
- Security management - security-related activities of information system management
- Transfer system - LCS and CN elements and communications protocols used by them and by ESs and RSs
- Physical and administrative environment - security related to environmental (physical) elements and personnel.

Figure 3-2 illustrates a generic view of several LSEs joined by CNs. Each LSE is defined and bounded by the elements under user (organization) control, including the environment. LSEs exhibit all or parts of each of the four generic architecture components, while the CN only represents a part of the transfer system.

End systems and relay systems are entirely contained within an LSE. Although Figure 3-2 shows ESs and RSs as separate generic components, in practice the same information system may combine both ES and RS functions as necessary. LSE connections to CNs are only through RS functions.

The security management component is not illustrated in this figure, but its functions are pervasive in the LSEs and extend to cooperate with CN management facilities.

The transfer system component is shown within dashed lines. Although it includes all of the LCS and CN elements, it includes only these portions of the ESs and RSs that implement communications protocols.

The physical and administrative environment component (labeled collectively as *environment* in Figure 3-2), represents all of the generic security services provided directly or indirectly by physical means (e.g., locked gates, guard dogs) or through administrative procedures (e.g., background investigations, issuance of badges).

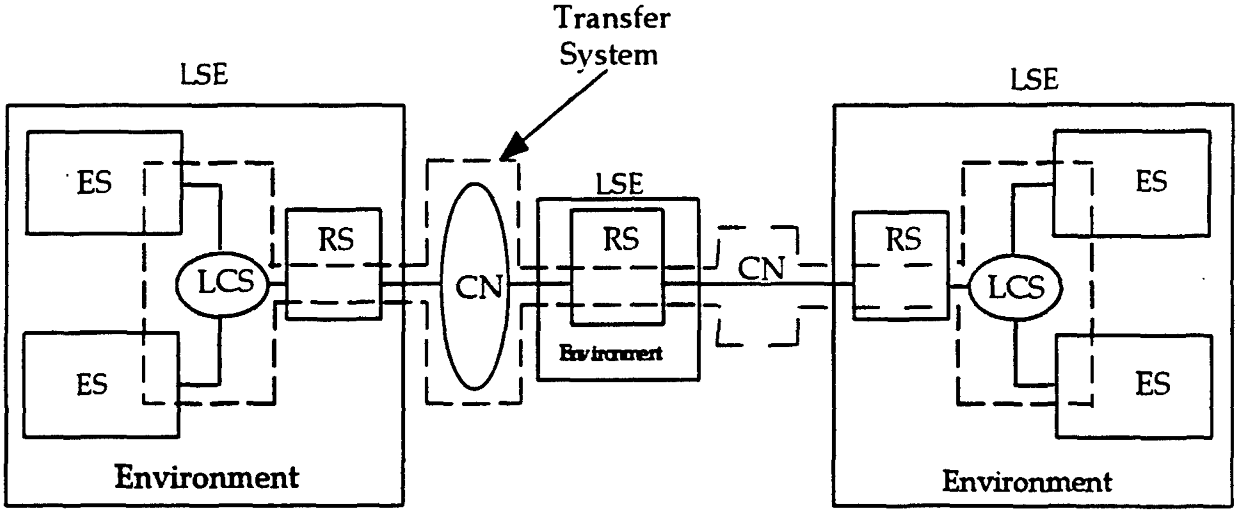


Figure 3-2. Generic Security Architecture View

3.2 SECURITY SERVICE ALLOCATIONS

The DGSA's security services are based on those defined in ISO 7498-2 (ISO, 1989a) for data communications. These security services include authentication, access control, data integrity, data confidentiality, and non-repudiation. (The *OSI Security Frameworks*, ISO 10181, is a multi-part standard that discusses each of these services, plus security audit and key management, in considerable detail.) In the DGSA, availability also is considered to be a basic security service. The basic security services are considered to apply not only to the transfer system, but are interpreted to apply to the entire LSE. This section discusses security service allocations to CNs and LSEs, and to the four generic security architecture components.

3.2.1 Abstract Architecture Security Service Allocations

In this section, security service allocations are made to the abstract security architecture components.

3.2.1.1 CN Security Service Allocation

In response to the requirements of Section 2, particularly the requirement to use common carrier services, the DGSA makes only a security service allocation of communications availability to CNs. CNs must provide an agreed level of responsiveness, continuity of service, and resistance to accidental and intentional threats to the communications service.

The reliability, flexibility, contingency actions, management, and preventive maintenance of CNs are some of the factors that will determine the availability of communications services. Protection of CN resources from accidental or intentional damage is both a security concern and, in the commercial world, a direct financial concern. Well-designed and well-managed CNs should exhibit graceful degradation in service and should provide for establishing priorities of service. CN providers will employ various security services to protect the CN's own resources to ensure that the agreed availability will be maintained. However, CNs are not relied upon for the confidentiality or integrity of the information they transfer. Failures in CNs can only result in the delay, misdelivery, or non-delivery of otherwise adequately protected information. The purpose of CN management, which is to counter these failures, is identical to that of the security service of availability.

3.2.1.2 LSE Security Service Allocations

All the security services are allocated to LSEs. The provision of security services for an entire LSE is accomplished by physical, administrative, and personnel security mechanisms. Physical LSE boundaries can limit facility access to authorized personnel. Protection of LSEs is provided in part by the logistical support system (e.g., configuration management control). In turn, LSEs provide protected environments for their end system, relay system, and LCS components. (See Section 7 for additional details.)

The open systems requirement of Section 2 demands that LSEs with highly sensitive information must have the ability to communicate with nonsecure as well as with secure LSEs. The architectural model for such LSEs is shown in Figures 3-3 and 3-4.

In Figure 3-3, a secure LSE is communicating with a nonsecure LSE that must be assumed to include hostile entities if the total information system is truly open. In this situation, no transfer system security services are used to protect information in transfer because none are needed and the nonsecure LSE offers no such services. The secure LSE must isolate its sensitive information (shown as shaded in the figure) and protect it with its own security mechanisms.

In Figure 3-4, both LSEs are considered secure (for at least some set of information) and cooperate to provide transfer system security services to protect the information in transfer. The secure LSEs must still protect themselves from nonsecure LSEs that are connected to the CN. The requirement for open systems provides serious challenges to the security architecture of LSEs.

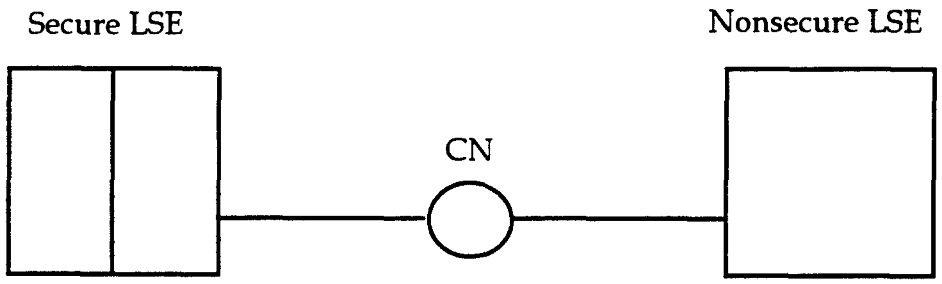


Figure 3-3. Secure-to-Nonsecure LSE Communications

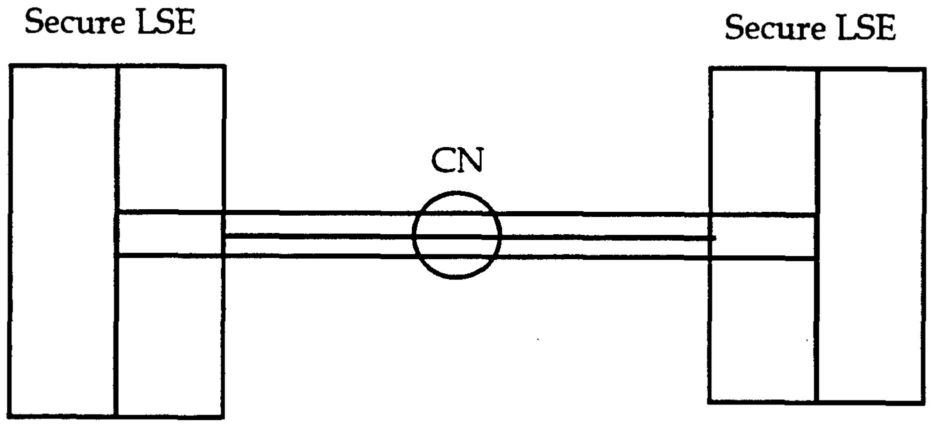


Figure 3-4. Secure LSE Communications

3.2.2 Generic Architecture Security Service Allocations

In this section, security service allocations are made to the generic security architecture components.

3.2.2.1 End System and Relay System Security Service Allocations

Security service allocations are made to end system and relay system hardware and software so that the hardware protects the software and the software protects information being processed, transferred, or stored. End system and relay system hardware and software collectively provide the security services of user identification and authentication, access control, data integrity, data confidentiality, non-repudiation, and availability. Details of the end system and relay system security architecture are discussed in Section 4.

3.2.2.2 Security Management Security Service Allocations

All the security services are allocated to the security management component, but only indirectly. The function of the security management component is to support and control the other architectural components. Security management applications and protocols are simply a portion of end system and relay system hardware and software compositions. Section 5 presents details of the security management architecture.

3.2.2.3 Transfer System Security Service Allocations

CNs have already been allocated the availability security service. LCSs are required only to provide the availability security service for communications among end systems and relay systems within LSEs. Other security services may be provided in LCSs for local purposes if they do not interfere with other requirements, such as interoperability with other LSEs.

Security services implemented within protected end systems and relay systems provide the basis for the protection of information being transferred. The remaining security service allocations to the transfer system make it responsible for peer entity and data origin authentication, access control, non-repudiation, confidentiality, integrity, and availability of information in transfer. The protection of information being transferred enables the protected distribution of security-relevant information for security management as well as user information. The sharing of identification and authentication information, audit records, key management information, and policy and privilege management information among LSEs can be safely accomplished if the transfer system is protected. Section 6 provides additional detail on the transfer system architecture.

There is a particular aspect of data confidentiality, usually referred to as *traffic flow security* (TFS), which is the responsibility of the transfer system. True TFS only can be provided by a class of security mechanisms that inherently conflict with some of the security policy statements (Section 2.3) upon which the DGSA is based. Under certain circumstances, it may be judged that the threats to a mission can only be countered using TFS. Because TFS mechanisms are costly and because some goals (e.g., interoperability) will be sacrificed to some degree, the employment of the TFS service must be carefully considered. See Section 6.3.1 for additional discussion of this topic.

3.2.2.4 Physical and Administrative Environment Security Service Allocations

All security services are allocated to the physical and administrative environment architecture component. Specific mechanisms to implement these services that protect the LSE are discussed in Section 7.

3.3 SECURITY CONCEPTS

The most significant capabilities of the DIS target architecture are distributed processing and open communications. The objectives for security in such an environment are to maintain open and distributed capabilities and yet be able to establish and enforce a wide range of mission and information security policies. A simple characterization of such an environment is that resources and information may be shared or isolated as desired.

The management of information is accomplished by individuals and groups of people who create, collect, process, categorize, store, transfer, and communicate particular information. The value of that information and, therefore, the required protection of that information is determined by the group. The group determines the conditions for authorized access to the information and the conditions for individuals to become members of the group. This approach applies equally to United States national classified information, trade secrets, proprietary data, or other identified collections of government, corporate or personal information. Three elements are necessary for this idea to be employed:

- A group must have a defined membership

- Information objects must be uniquely identified within the domain of the group

- The security policy regarding the protection of and access to the information objects must be known and agreed to by the membership.

Several concepts have been developed to support this approach to information management. They are information domains, strict isolation, and absolute protection. The ways in which these concepts influence and are supported by the DGSA generic architectural components (end systems and relay systems, security management, transfer system, and physical and administrative environment) are detailed in Sections 4, 5, 6, and 7.

3.3.1 Information Domains

An *information domain* is a set of users, their information objects, and a security policy. An information domain security policy is the statement of the criteria for membership in an information domain and the required protection of the information objects. Information domains are not hierarchically related, nor may they implicitly or explicitly infer a sensitivity relative to multiple categories of sensitivity.

In contrast to domains that might be composed of systems or networks, information domains are not bounded by systems or even networks of systems. Information domains are bounded by the presence of their identifiable information objects and may be supported by any information system that can meet the protection requirements of the information domain security policy. In this concept, a specific mission security policy may define several information domains, each with its own distinct information domain security policy. The security mechanisms of any number of information systems may be evaluated for their ability to meet these information domain security policies. Through the process of accreditation, these security mechanisms may be usable for part or all of one or more missions.

Each information domain is identified uniquely. The unique identification indicates (directly or indirectly) the sensitivity of all the information objects in an information domain. Any security-relevant attributes and attribute values of information objects in an information domain must be the same for all information objects in the information domain. That is, there must be no security-relevant distinction made among the information objects in an information domain. Members of an information domain may have different security-related attributes and attribute values. For example, some members might have only read permission for information objects in an information domain, while other members might have read and write permissions. Since all information objects in an information domain have the same security-relevant attributes and attribute values, a user who has read and write permissions in an information domain has those permissions for *every* information object in the information domain.

3.3.1.1 Interdomain Information Sharing and Transfer

Some mission requirements will necessitate the sharing or transfer of information objects among information domains. The establishment of new mission functions, new mission area relationships, or new organizations are examples of events that can create requirements for information sharing and transfer.

The simplest method of sharing information is to accept new members into an existing information domain and to grant access privileges to them. Where a need exists to share some, but not all, of the information objects in one or more information domains with members of other information domains, a new information domain may be created to contain the shared information objects. The new information domain, like any other information domain, requires a security policy. The members of the new information domain may or may not be members of the information domains from which its information objects were obtained.

Information objects can be transferred between two information domains only in accordance with established rules, conditions, and procedures expressed in the security policy of each of them. The transfer can be accomplished only by a user who is a member of both the sending and receiving information domains and, if required by the information domain policies, has been granted the appropriate privileges (e.g., "release authority").

The transfer of information objects between information domains may be implemented as a move operation (in which the information object no longer exists in the originating information domain), or as a copy operation (in which the information object exists in both information domains). Information objects moved or copied from one information domain to another must be relabeled with the label of the information domain to which the information object has been moved or copied.

In general, interdomain transfers can only occur within an end system or relay system. Interdomain transfers usually cannot occur among distributed end systems or relay systems; transfers among end systems or relay systems usually can only occur within the same information domain. These restrictions are consequences of the nature of security contexts and security associations that are used to create an appropriate environment for distributed information domain operations (see Section 6.1.3.2).

3.3.1.2 Security Contexts

A *security context* encompasses all end system resources and security mechanisms (including physical and administrative) that support the activity of a user operating in an information domain. When the end system ceases performing operations in one security context and begins performing operations in another, information cannot be allowed to pass from one security context to another unless a specific request is made. Also, communications among security contexts in an end system can only take place in accordance with the security policies of the information domains supported by the security contexts. Each information domain security policy must include a transfer policy which defines under what conditions information may move from one security context to another.

3.3.1.3 Security Associations

To support distributed processing, it is necessary to establish security contexts for the same information domain in the cooperating end systems. These contexts must communicate with one another with the same assurance as if they were in the same end system. A *security association* is the totality of communications and security mechanisms and functions (e.g., communications protocols, security protocols, security mechanisms and functions) that securely binds together two security contexts in different end systems or relay systems supporting the same information domain. A security association extends the protections required by an information domain security policy within an end system to information in transfer between two end systems. It also maintains strict isolation (see Section 3.3.2) from other information domains.

3.3.1.4 Multidomain Information Objects and Policies

The missions of most organizations require that their members operate in more than one information domain. The information management activities of a mission may be viewed as taking place in a set of information domains, some of which may be shared with other missions. To carry out their mission information management activities, users may need to process information objects from several information domains concurrently. Often, a user may have a *perception* that a collection of information objects from different information domains is a single, composite information object. Such a composite information object is referred to as a *multidomain information object.* This perception must be achieved without actually combining real information objects from different information domains to create real multidomain information objects. When creating the perception of multidomain information objects, strict isolation among information domains must be maintained, and the constituent information objects within the multidomain information object must be managed only in accordance with their individual information domain security policies. The purpose of multidomain information objects is to be able to define a collection of information objects to be displayed, printed, or transferred between information systems in a particular order or arrangement.

The creation and use of multidomain information objects must be subject to some security policy. The simplest policy is the one noted above, namely to conform to the policies of the individual information domains. However, in such cases it may not always be possible to print such a multidomain information object or to convey it to another user or information system. A multidomain information object security policy might be based upon some existing policy (e.g., U.S. national security policy) that states a relationship among the constituent information objects. Such a security policy for multidomain information objects is made part of the security policy of the information domains of the constituent information objects. In situations where the security policy for multidomain information objects is complex or involves several information domains, that security policy might be stated in one place in the supporting information system and be referred to by the individual information domain security policies.

Explicit multidomain information object security policies must state the specific privileges a user must have to view, print, create, delete, or transfer a multidomain information object between information systems. To create or otherwise deal with an entire multidomain information object, the user must be a member of each of the information domains in which the constituent parts of the multidomain information object are located. Some multidomain information object security policies might allow access only to the component parts of a multidomain information object for which the user has appropriate privileges, but in many cases this would not result in a sensible multidomain information object.

As noted in Section 3.3.1.1, information domains are not hierarchically related. Nonetheless, security policies for multidomain information objects may recognize marking rules that apply to the entire multidomain object or its parts based on existing policies (such as paragraph and page markings for information subject to U.S. national classification policy) when printed or displayed. Further, an information domain security policy is not precluded from recognizing that a user security clearance of Top Secret is adequate for access to the information objects in an information domain that contains U.S. national classification Secret information objects, if all other aspects of the information domain security policy are also met. (Note that the apparent hierarchy among U.S. national security policy classifications is actually a property of user privileges, in the form of clearances, rather than a relationship imposed on information of different classifications. Information that is classified Secret is *not* a subset of information that is classified Top Secret.)

The implementation of multidomain information objects in real information systems has many implications for end system, security management, and transfer system architectures. These implications are discussed further in Sections 5, 6, and 7.

3.3.2 Strict Isolation

The diversity of missions and the threats to the security of their information will result in information domain security policies with unrelated protection requirements. Thus, information systems that support multiple information domain security policies must adopt a protection strategy that provides a basis for satisfying all of them. One such strategy, termed *strict isolation*, is to isolate one information domain from another, except when there is an explicit relationship established. Under this strategy, an information system must provide mechanisms that maintain separation of information domains in ways that are satisfactory to each of them. The default information system security policy is strict isolation among the information domains supported.

In the absence of any information domain security policy to the contrary, an information object must be isolated. While such a situation is a logical possibility, in practice, all information objects should belong to an information domain that has a defined membership and an information domain security policy. Information domains with no explicit interdomain policies must adopt a policy of strict isolation to be enforced by the systems that support them.

3.3.3 Absolute Protection

Since open systems may consist of an unbounded number of unknown heterogeneous LSEs and it may be necessary to communicate with any of them, system security architects must have a rational basis for protection decisions in such an environment. In this environment, it is not possible to rely upon the assurances provided by physically separated networks or cryptographically isolated LSEs. Information domains must rely on the protections afforded by a heterogeneous collection of LSEs. The concept of *absolute protection* (which does not imply perfect protection) is set forth to provide a framework for achieving uniformity of protection in all information systems supporting a particular information domain. It directs its attention to the problems created by the interconnection of LSEs that provide disparate strengths of security protection.

In order to support an information domain in multiple LSEs, the overall strength of protection afforded to information objects must be consistent in those LSEs. Strength of protection is a function of the strength and correctness of security mechanisms (including physical and administrative environment) implemented in LSEs to satisfy an information domain security policy. The required strength of protection is determined by assessing the value of the information being protected and then assuming a hostile attacker has logical access to the LSE through the transfer system. *The specific mechanisms and their implementations need not be identical in every LSE that supports an information domain, but the implementations must provide at least the required strength of protection.*

If the overall strength of protection provided by each LSE supporting an information domain is successfully evaluated under the assumption that the LSE is logically accessible to a hostile user, then each of these LSEs can be accredited as being adequate to protect the information domain against the same threats. Protection provided in all the accredited LSEs under these conditions will be absolute, non-relative, and equivalent. Absolute protection is primarily concerned with the vulnerabilities created by connections to communications networks. This concept generally forces stronger mechanisms to be employed for information of a given sensitivity.

For system security architects, implementors, and accreditors to properly apply the concept of absolute protection, different approaches to evaluation of security mechanisms, components, and information systems will be required to determine equivalent protection. A single measure of overall strength of protection is not adequate. Rather, security mechanisms will need to be rated (measured) for their ability to support one or more security services, alone and in combination with other security mechanisms. The required strength of protection for an information domain will be translated to a set of such measures so that an appropriate set of security mechanisms can be chosen. This method of choosing security mechanisms will give security architects, implementors, and accreditors a consistent means for providing equivalent (though not necessarily identical) protection in the LSEs that support an information domain.

4.0 END SYSTEMS AND RELAY SYSTEMS

A generic security architecture for end systems and relay systems must be appropriate for a wide range of applications and environments. Among the many possible implementations, some unifying structure must be created that permits a generic approach to security. This structure must accommodate the requirements of Section 2 and the primary security allocations made in Section 3. This section refines several concepts presented in earlier sections for end system and relay system architectures, including security allocations, types of functions that are required to support the security allocations, types of devices that make up end systems and relay systems, and technologies that should be considered in specific implementations. Section 4.1 gives an overview of the end system security architecture, and its description is presented in Section 4.2. Section 4.3 lists candidate technologies to support implementations. Generally, relay systems provide services that require the same kinds of underlying support as end systems, except that they do not provide support for direct user interactions. Thus, a single security architecture for end systems and relays systems is appropriate. The remainder of this section refers to both end systems and relay systems simply as *end systems*.

Since the DGSA is a generic architecture, not all of its possible architectural choices and alternatives (security services and mechanisms) will be used in every specific implementation. The DGSA allows for a wide variety of specific implementations that will be dictated by missions and threats. Similarly, the generic end system security architecture must have wide applicability. The end system security architecture described here is a current best estimate of how the DGSA requirements can be met. To the extent that it depends on specific technological directions, it is subject to change as experience and technology dictate. However, the basic architectural decisions described should remain stable.

Much of the end system security architecture is similar to that proposed by Rushby (1984). There are some significant departures from Rushby's proposal, most notably with respect to centralization of security policy-related functions. Rushby argues for such functions to be tailored to and to be implemented with specific resource management functions. This argument is implicitly based on the fact that only a single, access control-based security policy is to be enforced. The DGSA requirement for supporting differing security policies per information domain (which may have other dimensions than simply access control) makes the argument for centralizing the basic security policy-related functions more attractive. More recent proposals for support of multiple security policies suggest architectural approaches which take a middle ground and may offer some performance advantages (see Abrams (1993) for a summary and extension of these approaches).

The end system security architecture focuses on conventional computer systems, which represent a large portion of all end systems. Other end system types may need to implement only portions of the end system security architecture. In extreme cases, such as simple sensor devices, the end system functions may be so limited that only specialized implementations of a small portion of the end system security architecture are appropriate (for example, such a device almost certainly would not need to support multiple information domain security policies).

4.1 END SYSTEM SECURITY ARCHITECTURE OVERVIEW

In Section 3, fundamental allocations of security services were made to LSEs and to the end systems and LCSs within LSEs. Security service allocations were made to LSEs to protect their resources, including end systems. The end system security architecture makes additional security service allocations to the end system hardware and software. Not every security service allocation needs to be made identically in every system. For example, if electronic emanations are considered to constitute a potential vulnerability, the responsibility for countering it could be assigned to the LSE or to one or more of its components. Similarly, there is flexibility with regard to how protection responsibilities are shared between end system hardware and software.

4.1.1 The LSE Protects the Hardware

As discussed in Section 3, the security service allocations to the LSE are implemented as physical and administrative security mechanisms. Administrative and environmental security mechanisms are discussed in more detail in Section 8. The primary security service allocations to the LSE are access control to facilities and some aspects of authentication of personnel. In addition, some aspects of information confidentiality and integrity, and system integrity and availability may be allocated to the LSE.

4.1.2 The Hardware Protects the Software

Section 3 assigned responsibilities to the end system for all security services. There are a variety of security mechanism choices available between the hardware and software portions of the end system, but certain general allocations and properties can be stated for the hardware.

The hardware is relied upon to function correctly, to enforce isolation of software functions, and to contribute to the protection of the integrity of the system applications and the operating system. It provides protected paths between users and trusted parts of the software. The hardware indirectly supports the isolation of information processed and stored in the end system by protecting the integrity of the software. Hardware mechanisms are used to protect the system from radio frequency interference and to prevent undesired emanations. In some environments, specific hardware technologies (e.g., protective coatings, hardened or alarmed containers) may be necessary to protect against tampering with end system components. Availability of an end system may be enhanced through technologies such as fault-tolerant and fault-detecting hardware features. Hardware cryptographic mechanisms are employed as needed to support various security services. Other hardware mechanisms (e.g., memory mapping) support specific aspects of the software architecture and are noted in the end system security architecture discussion (Section 4.2). There is an array of equipment available to support the hardware allocations.

4.1.3 The Software Protects Information

The security service allocations made to software are wide ranging. The portion of the transfer system supported by the end system software is responsible for the confidentiality and integrity of information transferred among end systems, for the authentication of end systems to one another, and for user authentication and access control in distributed systems. The details of how the transfer system is supported by end systems are presented in Section 6.

Security services and the mechanisms that implement them must be managed. The software applications that support security management in end systems are discussed in Section 5 and are extended in Section 6 for transfer system support.

The end system software is responsible for user authentication and access control, and for the integrity of information being processed and in storage. Correct operation of certain software is required to ensure end system availability. Additionally, the software is expected to provide functions that support the security policies and requirements stated in Section 2 that are not directly expressed as security services, such as support for multiple security policies. The remainder of this section refines the end system security architecture, which primarily is concerned with software structure.

4.2 END SYSTEM SECURITY ARCHITECTURE DESCRIPTION

A generic end system security architecture must respond to the security allocations discussed earlier, and it must be sufficiently flexible to encompass changing technology. The end system security architecture presented in Figure 4-1 is an example and not an implementation specification, and might be realized in several ways. The end system security architecture concentrates on support for multiple information domains with distinct security policies. Attention is paid to strict separation of information domains, management of end system resources, and controlled sharing and transfer of information among information domains. The end system security architecture also relies upon an engineering approach that seeks to isolate security-critical functions into relatively small modules that are related in well-defined ways. This approach has advantages in implementation, certification, and accreditation by limiting the scope of particular portions of these activities. While there are no existing end systems that specifically implement all of the end system security architecture, several efforts have been documented in the academic and research communities that support various aspects of the end system security architecture. Recently, commercial operating system vendors have adopted design and implementation strategies that share significant aspects of the end system security architecture.

A *security context* is a combination of all the LSE, hardware, system software, user application software, and information supporting the activities of a user (or system function) operating in an information domain. A security context builds on the common operating system notion of a user process space (sometimes called a *context*) as supported by hardware features and operating system functions. The primary distinctions between an ordinary user process space and a security context are that aspects of protection provided by the LSE are explicitly included, and that user applications operate in a controlled process space *subject to an information domain security policy*. Security contexts are described in more detail in Section 4.2.2.

A *separation kernel* manipulates the protection features of the end system hardware (e.g., processor state registers, memory mapping registers) to maintain strict separation among security contexts by creating separate address spaces for each of them. A separation kernel also controls communications among security contexts to allow sharing or transfer of information, and to allow services to be performed by one security context for another. All user security contexts and many system function security contexts are constrained to make requests for basic end system services on the separation kernel through a *standard kernel interface*. The separation kernel is described further in Section 4.2.1. The functions that make and enforce security policy decisions are intimately related to the separation kernel. These are described in Sections 4.2.1 and 4.2.3.1.

In Figure 4-1, end system software is divided into trusted (shown in the shaded area) and untrusted parts for practical evaluation. The trusted parts of the software are those that are considered so important to the secure operation of the end system that they must undergo strict evaluation procedures and come under strict configuration management control.

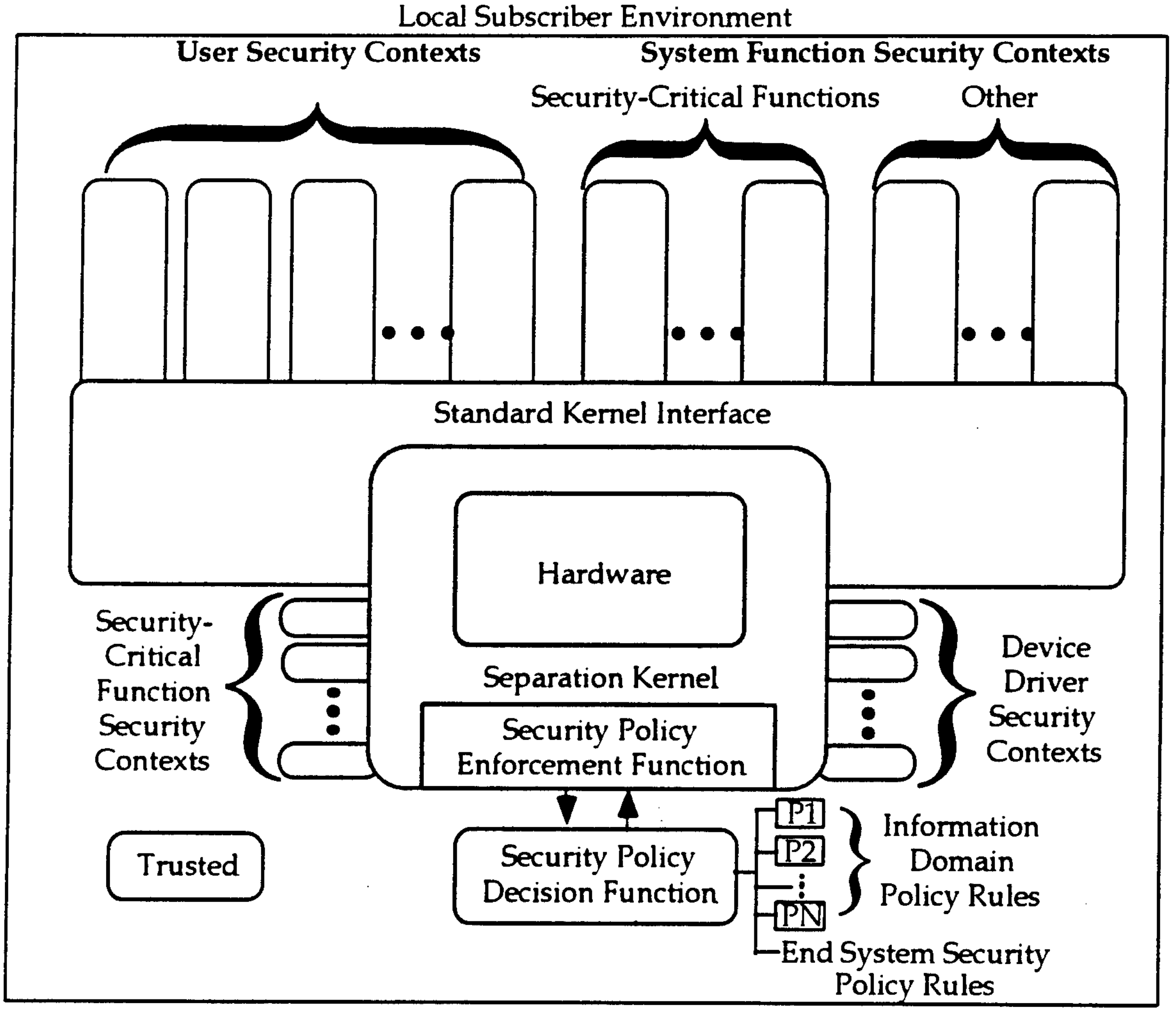

Local Subscriber Environment
User Security Contexts
System Function Security Contexts
Security-Critical Functions
Other
Standard Kernel Interface
Hardware
Separation Kernel
Security Policy Enforcement Function
Security-Critical Function Security Contexts
Device Driver Security Contexts
Trusted
Security Policy Decision Function
P1
P2
PN
Information Domain Policy Rules
End System Security Policy Rules

Figure 4-1. End System Security Architecture Generic View

The hardware (including any microcode) is considered trusted in the sense that its operation is assumed to be correct. Untrusted software is able to perform operations on basic system resources only through invocations of security-critical functions that are mediated by the separation kernel; inter-security context operations (e.g., inter-information domain communications) are performed by security-critical functions.

Untrusted *security-related* functions (such as security management applications and portions of transfer system applications) are expected to operate correctly to satisfy user operational needs, but need not be subjected to the rigorous scrutiny applied to the security-critical functions. Security-related software is not assumed to be free of security defects, although it is certainly prudent to obtain such software from reliable sources, test it before use, apply integrity safeguards to ensure it remains unchanged, and apply configuration management to it. (Software obtained from less than reliable sources may need to be inspected more carefully.) Under these conditions, if faulty application software is introduced into a system it will, at worst, prevent certain operations, but information compromise will not result because of the combination of strict isolation of information domains enforced by the end system, testing, and configuration management. The remaining software is not only untrusted, but is not expected to be examined for any security reasons.

The following subsections provide additional detail on the end system security software components, primarily for the separation kernel, security contexts, security-critical functions, and operating system implementations.

4.2.1 Separation Kernel

Much general operating system research has concentrated on organizing basic operating system functions into a collection called a kernel. The kernel presents abstractions of the fundamental resource management mechanisms to other, less primitive, service providers (information system functions and applications). In operating system implementations that attempt to provide a basis for secure information processing, the kernel software is carefully constructed and evaluated. To aid the evaluation process, the kernel functions are implemented as relatively small programs that are independent of one another to the maximum extent possible.

Rushby suggested that significant improvements in secure operating system kernel design and implementation could be achieved by isolating each kernel function in its own process space (i.e., address space). The benefit of this approach is that each operating system function performs a single, well-defined activity and can be understood and evaluated in relative isolation from all other functions. A separation kernel is charged with the critical task of providing separation among process spaces by manipulating the protection features of the end system hardware.

Until recently, most secure operating system designs have been limited with regard to security policy specification and enforcement. Particular limitations include support for only a single security policy (usually an access control policy) and the inability to change security policy conveniently. The end system security architecture adopts a particular view of operating system kernel design to meet DGSA requirements and concepts, most notably the support of multiple security policies so that a single end system can support users in different information domains simultaneously. The traditional operating system kernel functions are divided among the separation kernel, security policy enforcement and decision functions, and the remainder of the trusted operating system functions, called the *security-critical* functions. The separation kernel serves as the ultimate security policy enforcement function by mediating all use of the basic information system resources. The separation kernel notion is the foundation of the end system security architecture. However, any other information system mechanism that provides equivalent isolation of information domains and control of system resources is appropriate for implementations that are consistent with DGSA objectives.

The end system security architecture generalizes an approach that is becoming widely accepted concerning access control, namely the independence between the decision of whether or not an access to a resource is allowed and the enforcement of that decision. The separation of access control decision-making and access control enforcement functions allows the support of multiple access control policies. The ISO Access Control Framework (ISO, 1995d) designates these functions the *access control decision function* (ADF) and the *access control enforcement function* (AEF), respectively. In fact, most existing secure operating system designs have concerned themselves only with access control policy. Since one of the DGSA requirements is to support any security policy, the end system architecture extends the AEF concept to include the enforcement of all aspects of an information domain security policy. The resulting function is called the *security policy enforcement function* (SPEF). Similarly, the ADF concept is extended to a *security policy decision function* (SPDF). (The SPDF is discussed in more detail in Section 4.2.3.1.) The separation kernel is the implementation of the SPEF in the end system security architecture.

The separation kernel also is an extension (beyond access control) of the *reference validation mechanism* (RVM) described in the Trusted Computer System Evaluation Criteria (Department of Defense, 1985). The basic properties of the RVM must be applied to any separation kernel implementation: it must be invoked for every security-critical operation, it must be small enough to be verified, and its integrity must be maintained.

In the spirit of several current standardization efforts, a standard kernel interface will be defined to allow open system development of operating systems and applications built on implementations of the DGSA end system security architecture. The standard interface to the separation kernel is the same whether the underlying computer is a large multiprocessor mainframe or a single-processor workstation. This approach allows developers great latitude in implementing the separation kernel and the security-critical functions.

4.2.2 Security Contexts

From the perspective of the separation kernel, a security context is defined by a set of data and programs operating in accordance with an information domain security policy. As noted earlier, a security context also includes the physical and administrative security mechanisms of the LSE, and the hardware-based resources (e.g., registers, memory, disks) that are in use when the end system is serving a particular user (or system function). That is, a security context encompasses all end system resources and security mechanisms that support the activity of a user operating in an information domain. The separation kernel must maintain all the information needed to isolate one security context from another. When the end system ceases performing operations in one security context and begins performing operations in another security context, no information can be allowed to pass from one security context to the other unless a specific request is made and it is allowable under the security policies of the information domains involved.

Examples of information that end system security-critical functions (including the separation kernel) must maintain to support the operation and isolation of security contexts include:

- A unique identification for each security context
- The identification of the information domain being supported
- Hardware register values related to control of end system resources, including virtual memory and all devices in or attached to the end system
- The authenticated identity of the user being served
- The user's security attributes (permissions)
- Data structures needed to operate security-related functions and other untrusted system applications.

Each security context supports a user (or a system function) operating in a particular information domain. Over a period of time, an end system may maintain several security contexts to support one or more users operating in one or more information domains. A particular user might use (simultaneously or serially) security contexts operating in the same or different information domains. Different users may employ security contexts operating in the same or different information domains.

Since security contexts are isolated from one another by the separation kernel, communications among security contexts (requests for service or information transfer) in an end system can only take place in accordance with the security policies of the information domains supported by the security contexts. If the security policies of the supported information domains do not explicitly permit inter-information domain transfer, the SPDF will necessarily deny the request and the separation kernel will enforce that decision. Since an information domain contains the information of a particular user community, it would be unusual for an information domain security policy to prohibit information sharing between two security contexts supporting the same information domain.

Many end system activities are not carried out on behalf of a specific user (either an individual or the entire membership of an information domain as a group), but rather for basic end system operation and management. Examples of such activities include many of the security-critical system functions and end system management activities. These activities are carried out within end system security contexts on behalf of one or more of the information domains supported by the end system. The security policies of these end system information domains are created to exercise appropriate control of end system resources for all of the user information domains supported by the end system. Some example uses of end system information domains include the control and manipulation of multidomain objects, login applications, and management information domains.

Multidomain information objects (see Section 3.3.1.4) never exist in an end system except as displayed (or printed). Nonetheless, in end system implementations, it must be possible for a user to describe the relationships among the components of a multidomain information object so it can be displayed. Some implementations of multidomain information objects will result in the description being represented as an information object. Some security policies may preclude this information object from being held in any of the component information domains. In such cases, the end system must be able to create a system security context in which the description can be used by an appropriate application program that requests the display manager to construct the multidomain information object on a display device. Note that the multidomain information object description could be retained by the end system for future use by either the creator of the description or by other users who have the necessary information domain memberships. Similarly, the description could be transferred, in accordance with a multidomain object policy, (separately or with the component information objects) to another end system (see Section 6).

Before a security context can be created for the activities of a user in a particular information domain, the system must be informed which information domain is to be used. Ordinarily, the user's identity must be obtained and authenticated to determine if the user is a member of the requested information domain. One way of performing this startup function is to create a "login" security context that represents one of the end system information domains. The activities allowed in the login security context are limited to authenticating the user identity and starting a security context for the requested information domain (there might be a default information domain for a user recorded in the end system security management information base).

One useful resource control concept is *type enforcement*. The type enforcement concept generally restricts the input and output of a particular function to be of delineated types. In turn, the functions that are allowed to invoke other functions can be controlled by careful specification of input and output types. It is possible to impose a particular implementation of type enforcement by making specific security-critical functions "members" of particular end system information domains. Thus, only "member functions" of an end system information domain could invoke specific executable end system functions.

A consequence of the strict isolation aspects of the end system architecture is that many aspects of covert channels, both timing and storage, either cease to be concerns or are easily controlled. Possible storage channels are reduced to those between security contexts. If information domain policies are properly stated and the security policy, strict isolation, and interprocess communications functions are performing properly, there will be no covert storage channels available. To exploit timing channels between security contexts requires that a complete security context list is available so that a user can determine which security contexts (including end system security contexts) are in operation. Such information is part of one or more management information domains. It is not likely, and certainly not necessary, that an arbitrary user would be able to access such information. Even for those security contexts in which management information is available to its users, timing information for other security contexts should not be made available to those users.

4.2.3 Security-Critical Functions

The security-critical functions described in this section implement the various security services allocated to the end system and several additional supporting services. This set of security-critical functions is not necessarily complete as presented. Experience through prototyping and experimentation is needed to guide implementations that will meet all of the DGSA requirements, but the functions presented below should provide a sufficient basis for further research.

4.2.3.1 Security Policy Decision Function (SPDF)

The separation of security mechanisms from security policy enforcement and decisions is crucial to the flexibility of the end system security architecture. The SPDF is responsible for making all security policy decisions. The primary role of the SPDF is to isolate the rest of the end system software from knowledge of security policies. The importance of this approach is threefold.

First, the support of multiple information domains with different policies is accomplished easily because the security policies are represented in only one place and are interpreted by only one function. In many current secure system designs, it is difficult to point to the actual software code that implements the single security policy of those systems because it is embedded and scattered throughout code that performs multiple functions.

Second, by keeping security policy representations in one place, it is relatively easy to install, modify, or even replace the security policy for an information domain. It is not necessary to rewrite trusted software that implements the security policy. Rather, the rules that the SPDF interprets for an information domain are updated or replaced.

Third, changing the implementation of the SPDF would be transparent to the operation of the remainder of the end system software. Any correct implementation of the SPDF is acceptable, but it may be useful to standardize the representation of security attributes and security policy rules.

The SPDF approach will allow security-critical functions to be implemented independently of particular security policies. There is the potential in this approach that a computer vendor could support its entire customer base within a single end system software design. To illustrate this concept, consider an example of three enterprises with different, or even conflicting, security policies. The first is a DoD organization using a conventional DoD security policy. The second is a corporation with requirements for data integrity and data separation based solely on need-to-know authorization. The third is a university research laboratory that does not have any special security needs except a basic privacy-based access control policy. Without a policy-independent architecture, these three differing security policies would result in three different operating system implementations that could cause serious compatibility problems for a vendor trying to support all three environments. Using the SPDF approach, any or all of the three policies could be supported by the same end system software. If necessary, the three enterprises could be served by the same end system or (using the transfer system) they could share information as necessary across different end systems.

4.2.3.2 Authentication Function

The authentication function invokes one or more mechanisms used by an end system to identify and authenticate users (and to authenticate an end system to users), and for end systems to authenticate one another in a distributed environment. A common interface to the authentication function is used that is independent of the any information domain security policy or the authentication mechanisms employed. That is, the authentication function is the service interface to the mechanisms used to identify and authenticate users and end systems. The exact mechanisms selected will depend on the information domain policies in effect. An end system supporting multiple information domain policies may need to implement more than one authentication mechanism.

An authenticated user identity may be passed between information systems rather than the information used to authenticate that identity. That is, an end system supporting a particular information domain would be expected to accept that the authentication function has been performed reliably and correctly by other end systems supporting that information domain (use of the absolute protection concept makes this assumption reasonable). In some cases, it may be necessary to pass information about the authentication mechanisms used to validate the user identity. The transfer system is expected to protect the authenticated user identity as it is passed between information domains. Additional detail about distributed end system interactions is given in Section 7.

4.2.3.3 Audit Function

The audit function accepts audit messages from functions in the end system in accord with information domain and management information domain security policies. Audit records may become part of the security management information that is part of an information management domain (for one or more information domains or end system domains). Audit records may be directed to multiple repositories. In some cases, the audit information may best be used by an individual user (for example, time and method of most recent end system or information domain use). The audit function guarantees that audit messages cannot be lost and that the ordering of messages is preserved. As part of a distributed audit system, audit functions can forward the audit data they collect to a base-level, regional, or central audit center to alleviate local audit data storage requirements and to coordinate audit information from different end systems or LSEs. Audit data must be protected from unauthorized access or modification.

4.2.3.4 Process Scheduling Function

In operating systems that share the end system processor among multiple processes, the process scheduling function determines which of the processes next uses the processor (or processors in a multiprocessor end system) and for how long. The process scheduling function must be included among the security-critical functions so that no process can deny the processor to other processes either purposefully or inadvertently.

4.2.3.5 Device Management Functions and Device Controllers

The remainder of the security-critical functions are each responsible for a particular class of end system resources described below. These resources include memory, storage devices, display systems, interprocess communications, cryptographic services, and any other input/output devices controlled by the end system.

- The *memory management function* is responsible for controlling the use of memory by all software, including security-critical functions. It maintains memory-mapping information and controls the hardware functions that perform memory mapping.

- The *file management function* is responsible for controlling the use of storage media devices. Like the memory management function, it maintains disk-mapping (or other media-specific) information that provides basic virtualizations of the actual storage media. Other software (e.g., database programs) may build upon these virtualizations to provide even more abstract file structures to applications and users.

- The *display management function* is responsible for controlling the use of display devices (including screens and printers), keyboard devices, and pointing devices (e.g., trackballs, mice). The display management function provides basic display device operations. Because a single display device may be used to present information from multiple domains at the same time (typically through multiple windows or on paper), the display management function must maintain information that associates particular information to be displayed with the appropriate security context. Other software (e.g., an X Window System implementation) may provide requests to the display management function to achieve a particular display format.

- The *interprocess communications management function* is responsible for controlling the interprocess communications mechanisms (e.g., locks, semaphores, messages) used by all software processes in the end system. In particular, inter-context (e.g., inter-information domain) transfers are carried out through this function.

- The *cryptographic services management function* is responsible for controlling all of the cryptographically based security mechanisms in an end system. The security services it may support include confidentiality, data integrity, data origin authentication, and non-repudiation. The cryptographic management function may control a number of alternative cryptographic mechanisms to support different services and to provide different levels of protection that satisfy different security policies. The choice of mechanism may be based on many factors including the sensitivity of the data being protected, the security service requested, and the mechanisms available on other end systems for data that will be transferred.

- Each of the physical devices in the end system, including memory, disks and other storage devices, displays, cryptographic engines, specific user authentication devices, and communications interface controllers, has a corresponding software program that controls and passes information to and from it. These software programs collectively are called *device drivers*. Every device driver must be considered security critical because this software ultimately determines how a device operates. Although device drivers in older end system platforms were often quite large and complex, many contemporary devices contain much of the former device driver function in the device logic or in their own programs. Thus, many device drivers are now reasonably straightforward and follow well-known paradigms, which make their evaluation easier, although great reliance is placed on the correct implementation of the device.

4.2.4 Security-Related Functions

Some software functions within the end system are required to manage information or to provide an interface to the security-critical functions, but are not critical to system security. Of particular interest here are residual operating system functions, security management functions, and transfer system functions.

4.2.4.1 Residual Operating System Structure

Most of the security-critical functions are part of traditional operating system structures. Many other operating system components are not included in the security-critical functions, such as the user interface, utility functions, and high-level abstractions of information. These functions are present in varying forms in all traditional operating systems. The user interface, the particular utility functions, and the information abstractions provided characterize a particular operating system. That is, they distinguish one operating system from another even though they provide essentially the same services to a user. Because the security-critical functions provide commonly used, low-level services, many different operating systems can be implemented using them. Figure 4-2 is an abstract illustration of the software supporting a single security context.

Since security contexts are separated from one another, each can rely upon a different residual operating system structure. Thus, a single end system can support different operating system environments concurrently. Applications that were written to operate with a particular operating system should not require change unless they were allowed to directly manipulate basic operating system functions now controlled by security-critical functions.

Existing operating system implementations will need to be modified to use the standard kernel interface and the services provided by the security-critical functions. The degree of difficulty in making these modifications will be reduced if the original operating system implementation was

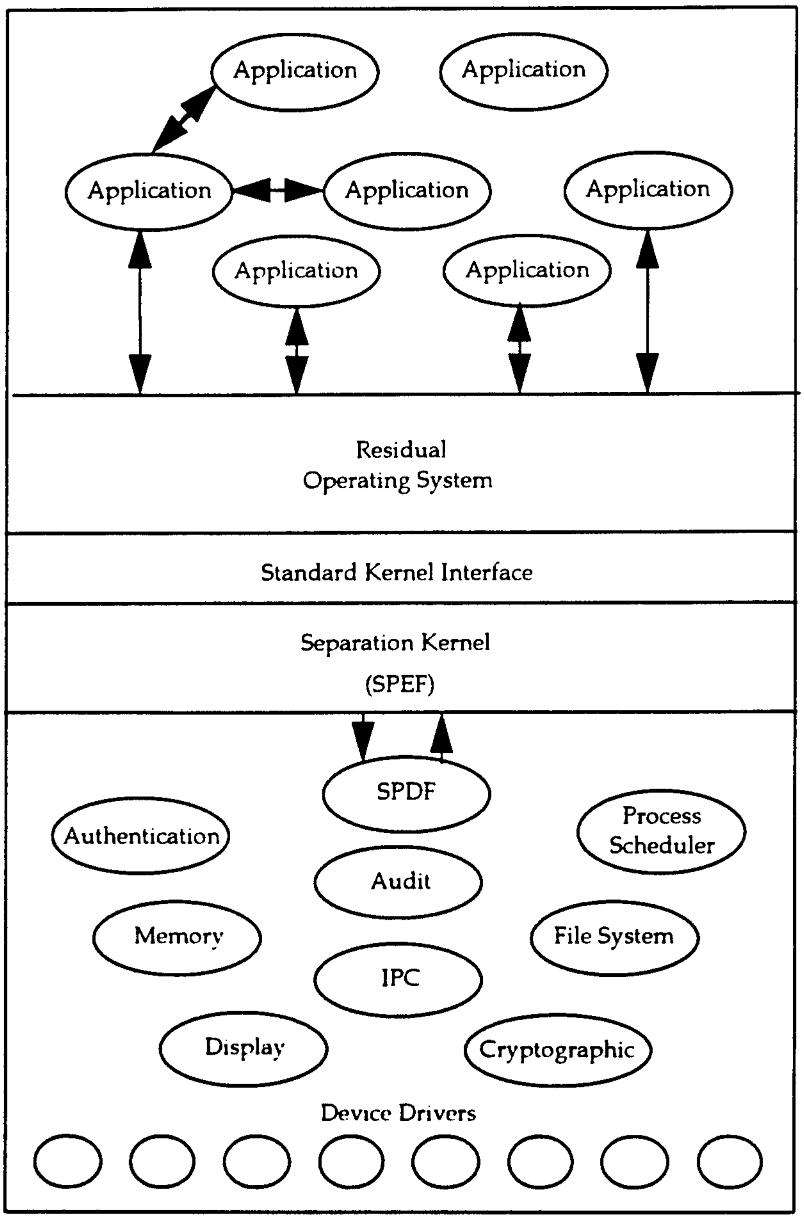


Figure 4-2. Security Context Software Component Relationships

well structured and modular. Some existing secure operating system implementations will adapt relatively easily to the use of the standard kernel interface, and many of the security-critical functions will already be present.

Residual operating system implementations structured to use the standard kernel interface to obtain basic services should be able to be moved among different hardware bases relatively easily since most hardware dependencies will be visible only in the separation kernel and the device drivers. This technique will enable applications to be used even in the face of changing hardware systems.

It should be noted that existing trusted software subsystems (e.g., trusted database applications) also will need to be restructured to fit the end system security architecture. It is possible that such a subsystem might be written to make direct use of the standard kernel interface (rather than calling on the residual operating system) for reasons of efficiency. It also is possible that existing trusted applications (which are appropriately structured) that run on dedicated servers may be able to support multiple information domains through carefully constructed interfaces.

4.2.4.2 Security Management Function

The primary role of the security management function is to control information needed by security-critical and security-related functions within the end system security architecture. Security management is a particular instance of general management functions. The concepts and structures defined in ISO 7498-2 and ISO 7498-4, have been adopted for use in the DGSA. Examples of the information manipulated by the security management function include information domain security policy rules used by the SPDF, configuration parameters for security mechanisms (e.g., cryptographic algorithms), configuration parameters for cryptographic mechanisms and end system devices, and audit information. Some information is managed for specific information domains and some is managed for end systems or LSEs. Details on security management are contained in Section 5.

4.2.4.3 Transfer System Function

The transfer system is defined in accordance with ISO 7498-1 and ISO 7498-2. Communications applications (e.g., X.400 electronic mail (International Telegraph and Telephone Consultative Committee (CCITT), 1988), X.500 directory services (CCITT, 1992), file transfer) and communications protocols used to communicate with other end systems are implemented as untrusted applications within the end system security architecture. These applications make requests for security services (which process information and generate protocol information) that provide required protection. For information to be transferred between end systems and within an information domain, a distributed security context is established through the use of security management and transfer system applications, and security-critical functions. Details of the transfer system are presented in Section 7.

4.3 END SYSTEM SECURITY ARCHITECTURE TECHNOLOGIES

Technologies that are considered to implement the end system security architecture affect all of the elements identified in Section 4.1 (local subscriber environment, hardware, and software).

4.3.1 LSE

The allocation of security services to the LSE requires that mechanisms must be in place to support those services. Physical and administrative security mechanisms will be used to implement the LSE protections. Some areas and mechanisms identified for additional investigation and research are biophysical (e.g., authentication, physical access control) and electronic (e.g., physical access control).

4.3.2 Hardware

The allocation of security services to the hardware requires that mechanisms must be in place to support those services. Some areas and mechanisms for additional investigation and research, and the security services they support, are listed below:

- Fault Tolerance - availability
- Fault Detection - availability, integrity
- Memory Management - strict isolation, integrity
- Protected Mode/Multistate Processors - strict isolation, integrity
- Majority Logic - availability, integrity
- Multiprocessor Architectures - availability, strict isolation, integrity
- TEMPEST - confidentiality
- QUADRANT - availability, integrity.

4.3.3 Software

The allocation of security services and other security-critical functions to the software requires that mechanisms must be in place to support those services. Some areas and mechanisms identified for additional investigation and research are listed below:

- Separation kernels - strict isolation and access control
    - Separation kernel interfaces
    - Process subsystems
        - -- Interprocess communications
        - -- Buffer caches
        - -- Security policy enforcement functions
- Security-critical functions - authentication, confidentiality, integrity, access control, non-repudiation
    - Security policy decision functions
    - Audit functions
    - Cryptographic engine functions

  - Device drivers
  - Window managers
- Security-relevant functions
  - Security management functions
  - Transfer system functions
- Trusted applications
  - Databases
  - X Window System
  - Operating systems.

5.0 SECURITY MANAGEMENT

Security management provides supporting services that contribute to the protection of information and resources in open systems in accordance with applicable information domain and information system security policies. This section builds on the definitions and concepts presented in Section 3.3. In Section 5.1, critical aspects of security management are related to architectural elements and concepts of the DGSA. In Section 5.2, clause 8 of ISO 7498-2 is used as the basis for presenting details of the DGSA security management architecture. Section 5.3 identifies tools needed by security architects and security administrators, and Section 5.4 discusses standards needed to support DGSA security management.

Security management is a particular instance of information system management. *Managed objects* are information system resources that may be managed. *Management information* is information associated with a managed object that is operated upon to manage that object. A human administrator employs a *management application process* (MAP) to use and maintain management information contained in a logical repository called a *management information base* (MIB). The contents of a single logical MIB may exist in several LSEs. When it is necessary to refer specifically to the processes and management information for security management, the terms *security MAP* (SMAP) and *security MIB* (SMIB) will be used. Otherwise, statements applying to MAPs and MIBs are understood to apply to SMAPs and SMIBs as well.

To ensure efficient and flexible system management, it is generally required that administrators have local or remote access to MIBs. As a result, MAPs will exist in all LSEs. CNs also contain MAPs and MIBs associated with their management. LSEs will manage their LCSs and also may need to cooperate with CN management. In most instances, this cooperation will not involve the use of security-related information since there are no shared security responsibilities.

Since management information comprises specially designated sets of information objects, these sets must exist within an information domain. Several possible choices can be made concerning the information domain in which particular management information objects exist relative to the information domain being managed:

- Each information domain may have a corresponding management information domain (1:1).
- A single management information domain may contain the management information objects for several information domains (1:many).
- The management information objects may be part of the information domain (embedded).

The first two choices are appropriate when the SMIB should not be contained in the information domain to be managed. The last choice, in which the MIB is a part of the information domain being managed, implies that every member of the information domain has the same access privileges to the MIB as to any other information objects in the information domain.

In addition, some management information objects may be associated with an entire information system and its functions. The system MIB might exist in its own management information domain or it might be placed in another management information domain (the latter situation is most likely when a "1:many" management information domain relationship is used).

5.1 SECURITY MANAGEMENT RELATIONSHIPS TO DGSA CONCEPTS

The requirement to manage multiple information domains has the most significant impact on traditional approaches to security management. Traditional security management is based on the assumption that all users of an end system are subject to the same security policy, so that a single view of security management is sufficient for the entire end system. End systems that support multiple information domains must provide the ability to manage each information domain independently. In addition, the use of security services and security mechanisms shared among multiple information domains requires security management coordination at the end system level. Thus, an end system security policy is necessary to specify how the shared use of security functions and resources among information domains is accomplished. This end system policy also must be managed.

As a result of this focus on security policy management, DGSA security management is mission driven and information oriented because information domains are the reflection of mission decisions on how to organize and control information. Section 2 discussed the relationships among missions, requirements, security policies, and security architectures, but only to the granularity of the entire mission. Information domains typically will reflect a major mission function, so further refinement of the mission-specific security policy into an information domain security policy will be necessary. It is not appropriate to specify exactly how that refinement should be done since only general guidelines exist for creating an information domain. However, a number of elements of information domain and end system security policies will be typical for a wide range of mission functions. Several of these security management elements of security policy are listed below, but the lists are not all-inclusive. Section 8 includes examples of incorporating security management policy elements into information domain and end system security policies.

A typical information domain security policy might include some or all of the types of information listed below. Not all of these information types will be reflected in the information domain security policy rules interpreted by the SPDF in an end system, but they are necessary to the development of those rules. Security management in end systems is concerned with the installation, maintenance, and enforcement of these rules and the information about users, security services, and security mechanisms needed to achieve a security policy. Not all security management activities are performed in end systems and relay systems. There are always supporting security management activities that are related to administrative and environmental security mechanisms or which are prerequisite to the use of end system security management functions (e.g., issuance of physical credentials to users, hiring and scheduling human guard services, or carrying out routine maintenance of physical barriers). Although these supporting activities are not called out in most parts of Section 5, they must be understood to be an integral part of security management. Examples of information domain security policy elements include:

- A brief description of the mission area and a more comprehensive description of the specific mission area function that the information domain supports
- A description of the information objects and their attributes, including rules pertaining to creation and use of multidomain information objects
- Membership criteria
- Rules for interdomain transfers, if any
- Security service requirements (including strength of service) appropriate to meet the risks determined by a threat analysis. Security services should be allocated to LSEs, end systems and relay systems, and the transfer system
- Criteria for acceptable security mechanisms to implement the required security services
- Security management-specific requirements
  - Relationship of the security management information domain to an information domain (1:1, 1:many, or embedded)
  - Criteria for security administrators (e.g., must be a member of the information domain, must not be a member of the information domain)
  - Roles, privileges, and duties of security administrators
  - Identities of security administrators
  - Configuration management requirements for the establishment or modification of information domain security policy rules
- Identification of one or more members of the information domain who are responsible for accrediting information systems that will support the information domain.

The security policy for an end system that supports multiple information domains must specify the management rules for conducting the following activities:

- Providing strict isolation among information domains
- Invoking and managing security mechanisms that implement the security services required by the security policies of the individual information domains
- Developing rules for the management of multidomain information objects, including criteria for user access, display labeling, and transfers between end systems

- Controlling and maintaining security management mechanisms and information objects that enable a security manager of a particular information domain to control that information domain independently of others.

The security policy rules for both end system security management and information domain security management are part of their SMIBs. For an information domain that is supported in more than one end system, the security administrator may have physical access to only some of those end systems. Thus, the SMAP that operates on the portion of a SMIB in a particular end system must be accessible to the security administrator both locally and remotely. A SMAP is like any other application in that it operates in a security context which represents a security administrator (or process) operating in a particular security management information domain. Thus, its security policy is interpreted and enforced by the SPDF and SPEF and it is subject to the same strict separation mechanisms as other information domains.

5.2 ISO 7498-2 AND DGSA SECURITY MANAGEMENT CONCEPTS

Clause 8 of ISO 7498-2 addresses many aspects of security management for open systems interconnection. The ISO 7498-2 security management structure is adopted as the basis for the DGSA security architecture and is extended to apply to all aspects of open systems security management.

5.2.1 Information Domains

ISO 7498-2 begins its security management discussion by considering security policy and security domains (clause 8.1.2):

> There can be many security policies imposed by the administration(s) of distributed open systems and OSI security management standards should support such policies. Entities that are subject to a single security policy, administered by a single authority, are sometimes collected into what has been called a "security domain". Security domains and their interactions are an important area for future extensions.

In the DGSA, "information domain" is substituted for "security domain." Some of the future extensions noted above have been included in the OSI Security Frameworks Overview, ISO 10181-1 (ISO, 1995c). The Frameworks Overview allows, but does not require, security domains to have subset and superset relationships. The DGSA does not allow information domains to be hierarchically related, and so has no need for the subset and superset notions. When sensitivity of information objects is a part of an information domain security policy, all the information objects in an information domain have the same sensitivity. The sensitivity of an information object is a consequence of its presence in an information domain. The "single authority" is the membership of an information domain. Usually the authority is delegated to one or more security administrators for day-to-day security management activities. The reference to "security domain...interactions" is accounted for in the DGSA by security policy interdomain transfer rules and their implementation.

5.2.2 Security Management Information Bases

ISO 7498-2 (clause 8.1.4) describes security management information bases as follows:

> The Security Management Information Base (SMIB) is the conceptual repository for all security-relevant information needed by open systems. This concept does not suggest any form for the storage of the information or its implementation. However, each end system must contain the necessary local information to enable it to enforce an appropriate security policy. The SMIB is a distributed information base to the extent that it is necessary to enforce a consistent security policy in a (logical or physical) grouping of end systems. In practice, parts of the SMIB may or may not be integrated with the MIB.

The DGSA uses SMIBs to conduct information domain and end system management, rather than for only end system management as implied above by the "appropriate security policy" for "each end system." As noted earlier, a distinct security management information domain may be responsible for the management of a single information domain (1:1) or several information domains (1:many), or the information domain may contain its security management information domain (embedded). The SMIB in these cases, respectively, contains security information for the single information domain, contains security information for all of the several information domains, or is contained in the information domain with its information objects. In the 1:many case, the information domains may or may not be related to the same mission. This flexibility allows a security administrator (or group of security administrators) to manage more than one information domain from the same SMIB. Also, it implies that each security administrator has the same attributes (privileges) with respect to the management information of all of the information domains that share a management information domain. (However, not every security administrator necessarily has the same attributes as the other security administrators.)

5.2.2.1 Information Domain SMIB Content

The following examples of information objects might be placed in a SMIB to manage an information domain:

- Information domain security policy rules
- Member registration information
- Member authentication criteria (e.g., strength of mechanism required)
- Member authentication information
- Member attributes (privileges) (e.g., access privileges, release authority for interdomain transfers)
- Visible security label information (i.e., what label, if any, is attached to information that is printed or displayed)

- Security service and security mechanism requirements for specific applications, including intradomain communications and interdomain information transfer.

5.2.2.2 End System SMIB Content

The end system SMIB contains information for management of security functions and resources shared by several information domains, including hardware resources, security-critical functions (particularly security services and mechanisms), and supporting applications (e.g., key management). More detail is given in later sections on several of the supporting security applications and related functions. The following example classes of information objects might be included in the end system SMIB:

- End system security policy rules

- Security services management information (see Section 5.2.7)

- Security mechanisms management information (see Section 5.2.8)

- Supporting services and mechanisms management information (e.g., alarm reporting, information system auditing, cryptographic key distribution, security contexts, security-critical functions, security-related applications operating for the end system).

5.2.3 Communication of Security Management Information

ISO 7498-2 (clause 8.1.5) observes the following about the communication of security management information:

> Management protocols, especially security management protocols, and the communication channels carrying the management information, are potentially vulnerable. Particular care must therefore be taken to ensure that the management protocols and information are protected such that the security protection provided for usual instances of communication is not weakened.

Security management information will be protected in accordance with the security policy of each management information domain. Management applications used to communicate security management information will rely upon the same open system protocol infrastructure as other applications. Management applications operate in security contexts. Security associations that ensure secure communications between security contexts in different end systems are described in Section 6.

5.2.4 Distributed Security Management Administration

ISO 7498-2 (clause 8.1.6) describes distributed security management administration:

> Security management may require the exchange of security-relevant information between various system administrations. in order that the SMIB can be established or extended. In some cases. the security-relevant information will be passed through non-OSI communication paths, and the local systems administrators will update the SMIB through methods not standardized by OSI. In other cases, it may be desirable to exchange such information over an OSI communication path in which case the information will be passed between two security management applications running in the real open systems. The security management application will use the communicated information to update the SMIB. Such updating of the SMIB may require the prior authorization of the appropriate security administrator.

The DGSA is consistent with this view and uses it as the basis for DGSA distributed security management. Each management information domain uses and maintains the SMIB for the information domain it manages. Security administrators may rely on a custodial infrastructure (e.g., communications security custodians). Cooperation with local administrators may be necessary for functions that cannot be managed remotely (e.g., aspects of key management that require physical access and personal accountability dictated by administrative and environmental considerations).

5.2.5 Security Management Application Protocols

ISO 7498-2 (clause 8.1.7) requires security management application protocols for exchange of security-relevant information:

> Application protocols will be defined for the exchange of security-relevant information over OSI communication channels.

There is not yet a clear preference among existing and developing security management application protocols. The general management application protocol defined by ISO is the Common Management Information Protocol (CMIP) (ISO, 1991). ISO also had defined the General Upper Layer Security (GULS) Security Exchange Service Element Protocol (SESEP) (ISO, 1994b). In addition, several security management functions have been defined with the series of standards within ISO 10164.

The Internet Engineering Task Force (IETF) defined the Simple Network Management Protocol (SNMP) (Case, 1989) and its successor, SNMP version 2 (Case, 1991). As the security management protocol situation becomes stable, the DGSA will adopt appropriate protocols.

5.2.6 End System Security Management Functions

ISO 7498-2 (clause 8.2.1) observes the following about system security management:

System security management is concerned with the management of security aspects of the overall OSI environment. The following list is typical of the activities which fall into this category of security management:

a) overall security policy management, including updates and maintenance of consistency;

b) interaction with other OSI management functions;

c) interaction with security service management and security mechanism management;

d) event handling management;

e) security audit management; and

f) security recovery management.

As noted previously, the DGSA broadens the view of end system security management to the entire open systems environment, especially with respect to the support of multiple information domains. The topics of event handling, security audit, and security recovery management are interrelated and will be treated together.

ISO 7498-2 (clause 8.3.1) describes event handling management as follows:

> The management aspects of event handling visible in OSI are the remote reporting of apparent attempts to violate system security and the modification of thresholds used to trigger event reporting.

ISO 7498-2 (clause 8.3.2) describes security audit management as follows:

> Security audit management may include:
>
> a) the selection of events to be logged and/or remotely collected;
>
> b) the enabling and disabling of audit trail logging of selected events;
>
> c) the remote collection of selected audit records; and,
>
> d) the preparation of security audit reports.

ISO 7498-2 (clause 8.3.3) describes security recovery management as follows:

> Security recovery management may include:
>
> a) maintenance of the rules used to react to real or suspected security violations;
>
> b) the remote reporting of apparent violations of system security; and
>
> c) security administrator interactions.

These security functions are related since the event handling function deals with all the apparent security violations recognized by an end system, the audit function selects those events that will be recorded, and the recovery function acts upon some of the selected events. The selection of audited events and those requiring a recovery action is determined by information domain security policies or by the end system security policy.

Event handling includes local as well as remote reporting of security-related events. Depending on whether a management entity (a security manager or a security recovery application) or a user is expected to examine or act on various alarms or audit records, alarm or audit information objects may be recorded in a particular management information domain SMIB, an end system SMIB, or a user-accessible file in an information domain.

Security recovery actions might include terminating a particular security context, temporarily prohibiting certain activities within an information domain, or disabling a particular communications interface. Some security recovery actions may depend on specialized data structures, such as a compromised cryptographic key material list, which controls continued use of key materials.

5.2.7 Security Service Management

ISO 7498-2 (clause 8.2.2) describes security service management as follows:

> Security service management is concerned with the management of particular security services. The following list is typical of the activities which may be performed in managing a particular security service:
>
> a) determination and assignment of the target security protection for the service;
>
> b) assignment and maintenance of rules for the selection (where alternatives exist) of the specific security mechanism to be employed to provide the requested security service;
>
> c) negotiation (locally and remotely) of available security mechanisms which require prior management agreement;
>
> d) invocation of specific security mechanisms via the appropriate security mechanism function, e.g., for the provision of administratively-imposed security services; and
>
> e) interaction with other security service management functions and security mechanism management functions.

An information domain security policy may be very specific about how security service requirements are to be met (by mandating particular security mechanisms). Alternatively, it may give only a general requirement for a security service of a particular strength and allow the SMAP to select an appropriate mechanism from those available. Each of the activities in the list above is concerned with an aspect of determining how security service requirements are satisfied by security mechanisms, as discussed below.

5.2.7.1 Determining and Assigning Strength of Service

Determining security services to be used and their strength is one aspect of developing a security policy for an information domain or an end system. The choices made are dependent on threats, vulnerabilities, and acceptable risk. That is, for large classes of information processing activities, a single determination of required security services can be made in advance because the value of the information being protected does not change often or quickly, nor do the vulnerabilities and risk. There are other classes of information activities for which it may be appropriate for a user to choose whether or not to employ a particular security service. For example, within the same information domain, some electronic mail messages may be of an informal or personal nature and not require a non-repudiation service, but other messages may be official business and may be required (by written policy) to employ a non-repudiation service. In cases like this, the user needs a selective means of invoking the security service, but the strength of the service is likely to be predetermined.

5.2.7.2 Assigning and Maintaining Rules for Mechanism Selection

For a given security service, one or more security mechanisms, alone or in combination with others, may be able to implement it. Some security mechanisms may be able to support more than one security service.

One of the aspects of the principle of absolute protection is that the security services chosen within an information domain security policy each have a minimum strength associated with them. Not all the security mechanisms that support a given security service need to be provided within end systems (or relay systems). In particular, the LSE may employ various administrative and environmental security mechanisms that contribute to the provision of one or more security services. As a result, the security mechanisms that support a given security service may be different when protecting information within an end system than when protecting information between end systems within the same LSE or between end systems in different LSEs. The resulting security service implementations must provide at least the minimum protection demanded by the security policy in all situations. Thus, to the extent that an end system supports security services with different mechanisms and a SMAP is aware (or can be made aware) of the distinctions among activities within an end system, between end systems in the same LSE, and between end systems in different LSEs, alternate choices of security mechanisms could be made.

The added complexity involved in making such choices might lead information system security architects to use only one set of mechanisms that satisfies an information domain security policy in all cases. However, in some situations this strategy would not be appropriate. For example, if some end systems in the same LSE often exchange large files, but only infrequently with end systems in different LSEs, a confidentiality mechanism necessary in the latter case might introduce an unacceptable performance penalty in the local situation, but administrative and environmental mechanisms could be relied upon to achieve the required level of protection.

5.2.7.3 Negotiating Available Security Mechanisms

One or more end systems that support the same information domain may be able to support a particular security service with more than one security mechanism, but it may not be known in advance of attempted communications which of these security mechanisms may be implemented in a specific end system. In such cases, the specific security mechanisms to be employed must be negotiated between the SMAPs in the end systems at the time the security association is established between them.

5.2.7.4 Invoking Security Mechanisms

The invocation of security services and security mechanisms within the end system security architecture involves several functions. Since all security services are security-critical, they are accessible only within the separation kernel, and applications can invoke them only through the standard kernel interface. Since most applications will rely upon the residual operating system for use of the standard kernel interface, the use of the interface will be transparent to those applications. If a request for a security service does not specify a security mechanism, the SMAP makes a choice among the available security mechanisms based on the information domain policy and invokes it through an appropriate operating system call. Otherwise, the SMAP invokes the specified security mechanism.

Although each application could make requests for security services and security mechanisms directly to the SMAP, there are significant advantages to adopting an Application Program Interface (API) approach. APIs provide a common set of subroutine calls to a related set of programming functions or services. An API not only relieves application designers of creating a specific set of interfaces, but also allows underlying services to be replaced (by equivalent mechanisms) without affecting the application implementation. Various efforts are defining APIs for the invocation of security mechanisms. One such effort is the General Security Service (GSS) API intended for use with the Internet suite of communications protocols (Linn, 1993). The GSS API and other related APIs could be used to invoke all security functions by making them the standard interfaces to the SMAP (they could be incorporated into the SMAP). GULS provides a standard set of protocol elements that can be used by applications to convey protected information between end systems.

The use of a combination of the GSS API, GULS, SMAPs, and the standard kernel interface can contribute to the independence of security services and security mechanisms and to their transparency to users and applications. This independence allows different security mechanisms to be accommodated at various stages in an end system life cycle, and for end systems to accommodate information domains with different security service requirements.

5.2.7.5 Specifying Interactions Among Security Service and Mechanism Management Functions

The use of some security services depends on the results of others. For example, access control usually employs the output of the authentication service. Required security service interactions must be expressed in a security policy. Similarly, some security mechanisms are dependent on others or on supporting security functions, for example, key management for cryptographic security mechanisms. These dependencies must be part of the SMIB so the SMAP can invoke the appropriate security mechanisms and functions.

5.2.8 Security Mechanism Management

ISO 7498-2 (clause 8.2.3) describes security mechanism management as follows:

> Security mechanism management is concerned with the management of particular security mechanisms. The following list of security mechanism management functions is typical but not exhaustive:
>
> a) key management;
>
> b) encipherment management;
>
> c) digital signature management;
>
> d) access control management;
>
> e) data integrity management;
>
> f) authentication management;
>
> g) traffic padding management;
>
> h) routing control management; and,
>
> i) notarization management.

The DGSA adopts this list and adds availability management.

5.2.8.1 Key Management

ISO 7498-2 (clause 8.4.1) describes key management as follows:

> Key management may involve:
>
> a) generating suitable keys at intervals commensurate with the level of security required;
>
> b) determining, in accordance with access control requirements, of which entities should receive a copy of each key; and,
>
> c) making available or distributing the keys in a secure manner to entity instances in real open systems.
>
> It is understood that some key management functions will be performed outside the OSI environment. These include the physical distribution of keys by trusted means.

> Exchange of working keys for use during an association is a normal layer protocol function. Selection of working keys may also be accomplished by access to a key distribution center or by pre-distribution via management protocols."

The DGSA relies upon standard key management techniques. Specifically, a Security Association Management Protocol (SAMP) is a necessary part of the transfer system. There are several competing SAMP developments in progress. Among them is the Institute of Electrical and Electronics Engineers (IEEE) 802.10 Standard for Interoperable LAN/MAN Security (SILS) Part 3 (IEEE, 1995), which has recently become the basis for the key management protocol standard being developed in ISO. The IETF is considering several alternative proposals. The DGSA requires a SAMP that will be sufficiently general to support security association establishment as described in Section 6.

There is an evolving key distribution system for U.S. Government use, the Electronic Key Management System (EKMS), from which the majority of U.S. Government cryptographic keying materials are generated and distributed. The EKMS Local Management Device (LMD) is the EKMS presence in LSEs. The EKMS is adopted as part of DGSA guidance. Although this is specific guidance, it is necessary because key management and cryptographic systems are being developed independently by vendors. A potential customer might procure several key management devices just to support a large, base-level LSE, some of which could be based on proprietary security management systems for vendor-specific end systems or LCS security products. These key management systems would almost certainly be incompatible with one another, thus increasing both initial and life-cycle costs, and impeding interoperability. The clear long-term solution is to develop key management and cryptographic products (including the evolving EKMS) based on the forthcoming standards.

5.2.8.2 Encipherment Management

ISO 7498-2 (clause 8.4.2) describes encipherment management as follows:

> Encipherment management may involve:
>
> a) interaction with key management;
>
> b) establishment of cryptographic parameters; and,
>
> c) cryptographic synchronization.
>
> The existence of an encipherment mechanism implies the use of key management and of common ways to reference the cryptographic algorithms.
>
> The degree of discrimination of protection afforded by encipherment is determined by which entities within the OSI environment are independently keyed. This is in turn determined, in general, by the security architecture and specifically by the key management mechanism.
>
> A common reference for cryptographic algorithms can be obtained by using a register for cryptographic algorithms or by prior agreements between entities.

It is expected that new cryptographic products will support multiple algorithms that can be selected by each application. In such an environment, the registration of cryptographic algorithms will be necessary so that algorithm selection can be negotiated between end systems. The ability to select a cryptographic algorithm has implications for the security management of the devices involved, such as determining under what conditions an algorithm can be employed and for auditing algorithm use.

5.2.8.3 Digital Signature Management

ISO 7498-2 (clause 8.4.3) describes digital signature management as follows:

> Digital signature management may involve:
>
> a) interaction with key management;
>
> b) establishment of cryptographic parameters and algorithms; and
>
> c) use of protocol between communicating entities and possibly a third party.
>
> Note: Generally, there exist strong similarities between digital signature management and encipherment management.

When digital signatures support a non-repudiation service that relies upon a trusted third party, additional security management responsibilities may be added with respect to long-term archiving of keys and algorithm identifiers so that transactions can be verified well after they occur.

5.2.8.4 Access Control Management

ISO 7498-2 (clause 8.4.4) describes access control management as follows:

> Access control management may involve distribution of security attributes (including passwords) or updates to access control lists or capabilities lists. It may also involve the use of a protocol between communication entities and other entities providing access control services.

The "distribution of security attributes" includes their initial installation in a SMIB. Since not all the information in an information domain SMIB is necessarily locally present in every end system that supports an information domain, it may be necessary to convey access control attributes between end systems. Note that user-specific access control attributes may not always be required since an information domain security policy may confer certain access rights on all its members.

5.2.8.5 Data Integrity Management

ISO 7498-2 (clause 8.4.5) describes data integrity management as follows:

> Data integrity management may involve:
>
> a) interaction with key management;
>
> b) establishment of cryptographic parameters and algorithms; and,
>
> c) use of protocol between communicating entities.

When using cryptographic techniques to support the data integrity service, similarities exist between data integrity management and encipherment management. In some instances, within a single end system, data integrity can be attained as a by-product of strong access control mechanisms. When a strong communications data integrity service is required, cryptographic mechanisms are likely candidates. A SAMP must provide means for selecting algorithms and keys for data integrity.

5.2.8.6 Authentication Management

ISO 7498-2 (clause 8.4.6) describes authentication management as follows:

> Authentication management may involve distribution of descriptive information, passwords or keys (using key management) to entities required to perform authentication. It may also involve use of a protocol between communicating entities and other entities providing authentication services.

Authentication mechanisms rely upon particular authentication information to validate a given identity. The authentication information against which user-supplied authentication information is verified is stored in the SMIB and is subject to similar considerations as access control attributes. It should be noted that an authenticated individual identity may not be required by some information domain policies since it may be sufficient that an individual has been physically identified and allowed access to an end system to assert membership in an information domain.

5.2.8.7 Traffic Padding Management

ISO 7498-2 (clause 8.4.7) describes traffic padding management as follows:

> Traffic padding management may include maintenance of the rules to be used for traffic padding. For example, this may include:
>
> a) pre-specified data rates;
>
> b) specifying random data rates;
>
> c) specifying message characteristics such as length; and
>
> d) variation of the specification, possibly in accordance with time of day and/or calendar.

Traffic padding in physical layer communications devices is often managed as a configuration parameter. In an open systems environment, traffic padding in the physical layer will occur infrequently. Traffic padding in application layer protocols could be invoked as the result of a user request or as the result of an information domain security policy requirement applied to all or some class of communications. The critical management aspect of satisfying such a request is to assure that the padding is applied at the correct stage of processing with respect to other security services, such as data integrity or data confidentiality.

5.2.8.8 Routing Control Management

ISO 7498-2 (clause 8.4.8) defines routing control management as follows.

> Routing control management may involve the definition of the links or sub-networks which are considered to be either secured or trusted with respect to particular criteria.

Routing control in open systems meeting DGSA requirements will normally be restricted to choosing a particular network interface when an end system is connected to multiple CNs or LCSs.

5.2.8.9 Notarization Management

ISO 7498-2 (clause 8.4.2) defines notarization management as follows.

> Notarization management may include:
>
> a) the distribution of information about notaries;
>
> b) the use of a protocol between a notary and the communicating entities; and
>
> c) interaction with notaries.

See Section 5.2.8.3.

5.2.8.10 Availability Management

Availability management is not described in ISO 7498-2. Availability management is limited to interactions with the LCS- or CN-provided management facilities for notifications of outages and, if applicable, alternate service information.

5.3 SECURITY MANAGEMENT TOOLS

Security architects will need various tools to enable them to design end systems that will support user requirements as reflected in information domain security policies. Security administrators must have available a set of tools to assist them in performing their functions efficiently and conveniently. Not all of the tools discussed here are available currently, and steps will need to be taken to ensure their timely creation.

5.3.1 Security Policy Rule Specification

To complement the development of the SPDF, a tool must be developed to assist in or perform the reduction of security policies to security policy rules that can be interpreted by the SPDF. The specification of security policy rules is a new endeavor and will require a significant research effort.

5.3.2 Security Mechanisms Catalog

The selection of appropriate security mechanisms to implement the security services required by security policies is an activity that will require specific support that does not yet exist. There are several interrelated factors that must be considered.

The first factor is the strength of security mechanisms and other security-critical functions (e.g., separation kernel effectiveness). The second factor is the characteristics of security mechanisms, that is, what they do and do not provide, how security mechanisms interact with one another, and implementation and employment requirements for security mechanisms to work effectively. The third factor is the cost of security mechanisms, including both procurement and life-cycle costs (to include supporting functions such as key distribution). The fourth factor is user impacts, such as performance penalties.

To an extent, some of these factors are considered in current procedures for evaluating security products. To support security architects in suggesting appropriate security mechanism choices, all of these factors must be considered. Evaluations based on these factors could be performed on implementations of particular security mechanisms or on products that implement multiple security mechanisms. The result of such evaluations would be a security mechanisms and product catalog from which security architects could make appropriate choices.

One significant aspect of the evaluations for such a catalog is that they would not result in a single composite rating for a security mechanism or product. Each security mechanism would be rated for its strength in support of a particular security service. A security mechanism that supports more than one security service would have more than one strength rating. The security mechanism might have a different strength rating when used in conjunction with one security mechanism than it would with another. A security product would have strength ratings for each of its mechanisms. Clearly, establishing metrics for these strength ratings will be a formidable and critical aspect of creating the catalog.

5.3.3 Maintenance Applications for Security Administrators

Each of the security management activities discussed in Section 5.2 will require automated support for security administrators. The applications that provide this support are concerned with various aspects of SMIB maintenance, key management, and examination, processing, and correlation of information such as audit records. These management applications should work together smoothly, but they must also be separable if it is desired to assign certain activities to specific security administrators. In some instances, it will be necessary to integrate security management applications with other applications. For example, X.500 Directory Service Agents might be used to store portions of a SMIB so that user certificates are easily available to a user community.

5.4 AREAS FOR SECURITY MANAGEMENT STANDARDIZATION

Standardization of security management functions, data structures, and protocols will enable interoperation of SMAPs across many end system platforms and, thus, allow effective distributed security management. Areas for security management standardization include, but are not limited to the following:

- Security policy rule representations so that security policies can be installed remotely
- Key management functions that support the generation, distribution, and accounting of cryptographic key material
- Audit information formats so security management applications can interpret events occurring on multiple end systems that support multiple security domains
- Protocols for the exchange of security management information and for remote security management operations.

6.0 TRANSFER SYSTEM

This section discusses the basic goal of the transfer system security architecture and then the means to achieve that goal. Section 6.1 discusses the basic notion of distributed security contexts and the primary function that supports them, the security association. Section 6.2 describes several supporting functions and tools needed to implement distributed security contexts and security associations. Section 6.3 discusses the relationship of the transfer system security architecture to some specific security-related topics.

In Section 3, the transfer system was identified as the LCSs, CNs, and the communications protocols in end systems and relay systems. Security services allocated to the transfer system provide the basis for the protection of information in transfer. Availability is the only security service allocated to CNs and LCSs. Additional security services may be provided by LCSs, but they are only applicable to local communications.

The portion of the transfer system in end systems and relay systems consists of open system networking applications and communications protocols (including some security protocols). These applications and protocols are executed in the same security context as other user applications for a user operating in a particular information domain. Except for transfer system functions that are among the security-critical functions (e.g., network interface device drivers, cryptographic functions), transfer system software does not need to be trusted. The transfer system must be managed, so the SMAP and SMIB of Section 6 are extended to account for transfer system functions.

The primary goal of the transfer system security architecture is to provide protection of information in transfer to support information sharing and distributed processing within the security architectures of the other DGSA elements and the fundamental concepts. The basic approach to achieving this goal is to enable security contexts in different end systems or relay systems (that support the same information domain) to communicate as if they were in the same end system or relay system. The transfer system security architecture must fit within the end system and relay system architecture of Section 4 and the security management architecture of Section 5, and it must extend the support of fundamental DGSA concepts to communications, especially information domains, strict isolation, multidomain information objects, and absolute protection. The remainder of Section 6 addresses various concepts and functions needed for achieving the transfer system goal.

6.1 DISTRIBUTED SECURITY CONTEXTS

The generic transfer system security architecture seeks to create structures in which applications in security contexts in different end systems or relay systems (that support the same information domain) communicate with the same assurance as if they were in the same end system or relay system. Such structures are referred to as *distributed security contexts*. There are two basic classes of communications that must be considered, *interactive* and *staged delivery*. Staged delivery refers to communications in which the information being transferred is sent from the originating end system application to a relay system application, in its entirety, and then is sent from the relay system application to the destination end system application. (There may be several relay system applications involved before the information is finally delivered to the destination end system application.) The most common example of staged delivery is electronic mail. Interactive communications include all non-staged delivery applications. The means used to create distributed security contexts are different for interactive and staged delivery communications and will be discussed separately.

6.1.1 Distributed Security Contexts for Interactive Communications

An *interactive distributed security context* is formed when two security contexts in different end systems are joined securely using a set of mechanisms that is referred to as a security association. A *security association* is the totality of communications and security mechanisms and functions (e.g., communications protocols, security protocols, administrative and environmental security mechanisms, security-critical mechanisms and functions) that securely binds together two security contexts in different end systems or relay systems supporting the same information domain[1]. A security association extends the protections required by an information domain security policy within an end system to information in transfer between two end systems and it maintains strict isolation from other information domains. A security association can be considered an extension or expansion of an OSI application association. OSI application layer entities in different end systems employ application associations to communicate. An application association is composed of appropriate application layer functions and protocols plus all of the underlying communications functions and protocols at other layers. A security association is an application association that includes additional support from security functions and mechanisms. The security management information for a security association is contained in a SMIB and includes all the security-relevant attributes required to establish and maintain a security association, such as the information domain label and secure communications attributes (e.g., cryptographic algorithm identifiers and keys).

Making a decision about whether to allow establishment of a security association may require several related functions to be performed such as the exchange and processing of security attributes of the user (e.g., authenticated identity, access privileges). These attributes might be contained in a security certificate such as that defined in the X.509 Directory Services Authentication Framework (CCITT, 1992). The information contained in an X.509 certificate may be signed by any number of hierarchically related certificate-issuing authorities, down to an information domain-specific certificate-issuing authority if that level of granularity is required. This signature verification adds greater assurance to the credibility of the information contained in the certificate.

[1] Note that the DGSA meanings of security association and security association management protocol are more general than their meanings in existing protocol specifications.

Multiple security protocols may be included in a single security association to provide a combination of security services. For example, a network layer protocol might provide continuous end system origin authentication and data integrity, while a presentation layer protocol might provide selective field data confidentiality. Some lower layer security protocols can multiplex several security associations between the same end systems. The security associations share the same cryptographic algorithm and keys. This arrangement may be appropriate for interactive distributed security contexts that support the same information domain, but it is unlikely to be acceptable for different information domains because of strict isolation requirements.

In some instances, an interactive distributed security context will be formed between end systems that employ no security protocols and may not even require an authenticated user identity. Such instances include access to public information utilities (e.g., a news wire service feed) or completely unprotected end systems. In these instances, an end system that supports other information domains, will be entirely responsible for maintaining the isolation of unprotected information domains from other information domains.

Some communications between end systems involve information that is not ordinarily stored in an end system, for example, real-time voice and video applications. In these cases, users must monitor and enforce the accuracy of the security context and association established for the distributed security context. That is, humans must ensure that information exchanged belongs to the information domain represented by the distributed security context as is currently done when using Secure Telephone Unit-IIIs for secure voice or data communications.

6.1.2 Staged Delivery Distributed Security Contexts

A staged delivery distributed security context is transferred from the originating end system to the destination end system. This is accomplished by an application in the originating end system cryptographically wrapping the information to be transferred in a form that allows the destination end system to reconstitute the security context in which the information was wrapped. The wrapped information is transferred (in stages) from the originating end system to the destination end system. Ideally, the wrapping process should provide all security protection of the information while in transfer. No security services (other than availability) should be expected of the application relay systems involved in the staged delivery because they might be provided by common carrier providers, as is the case for CNs. If the wrapping process cannot provide all the necessary security protection, the application relay systems will have to be implemented to support the DGSA and interactive distributed security contexts between end systems and relay systems will have to be used to ensure the secure staged transfer of information.

There is an existing specification for a secure electronic mail service that satisfies the requirements for staged relay distributed security contexts. This document is the Secure Data Network System (SDNS) Message Security Protocol (MSP) specification (NSA, 1992). For details of how secure staged delivery can be achieved, the MSP specification should be examined. MSP will be the basis for secure messaging in DoD as Phase II of the DMS is implemented and deployed.

6.1.3 Other Aspects of Distributed Security Contexts

This section provides additional discussion of two specific aspects of distributed security contexts.

6.1.3.1 Multidomain Object Transfer

Section 3.3.1.4 defined and discussed multidomain objects and noted that their purpose is to display or print related information objects from several information domains in an ordered format. Section 3.2.2 discussed some high-level implementation aspects of multidomain objects. The transfer of a multidomain object between end systems requires that both the component information objects and the description of their relationships be transferred. Since a distributed security context supports transfer of information within a single information domain, one distributed security context is used for each of the component information domains. If the description of the component relationships is contained in an information object in a separate information domain, another distributed security context is required for its transfer. An application similar to those used to display or print multidomain objects is needed to coordinate the transfer of the component information objects.

6.1.3.2 Distributed Security Context Single Information Domain Restriction

The definition of a distributed security context restricts it to joining end system or relay system security contexts that support the same information domain. In principle, this restriction could be removed under some conditions for some information domain security policies, however, there are practical reasons for retaining it. One of the principal functions of a distributed security context is to maintain strict isolation of information in transfer. Within an end system, the separation kernel (or other strict isolation mechanism) controls all interactions between security contexts. As noted earlier, it is expected that cryptographic mechanisms will be the usual means to maintain strict isolation for information in transfer. The use of such cryptographic mechanisms requires shared use of keys and other supporting information between security contexts in the communicating end systems. If those security contexts support different information domains, sharing of the keying information is difficult. There will also be additional complexity introduced into many communications and security protocols that will result in trusted implementation of additional functions. The restriction that distributed security contexts support transfers within a single information domain is intended to simplify implementations that support the DGSA concepts.

6.2 TRANSFER SYSTEM SUPPORT

This section describes several elements needed to support the basic transfer system activities.

6.2.1 Security Management Application Process

In addition to the SMAP functions described in Section 5, the SMAP also controls the establishment and termination of all security associations and distributed security contexts, and all transfer system security services and mechanisms. Additional transfer system-related SMAP functions and interfaces support the following activities:

- End system communications applications requests (e.g., through the GSS-API)
- Additional SMIB information object use and maintenance (e.g., to access information for remote security administration maintenance, security protocol and algorithm operation, certificate processing)
- Maintenance and retrieval of security information from the X.500 Directory using the directory access protocol
- MSP processing for staged delivery secure messaging for both transmission and receipt
- SAMP operations for establishment of interactive distributed security contexts, including security protocol operation, termination, and recovery, plus maintenance of SMIB entries for each security association established
- General-purpose management protocol operation (e.g., CMIP) to accomplish secure exchange of security information between distributed SMAPs or network management information requested by network management systems.

6.2.2 Security Management Information Base

Additional information is required in the end system SMIB and the information domain SMIBs to support transfer system operations.

Additional information domain SMIB information items include:

- X.509 certificates to carry appropriate security information, such as key management certificates
- User access control information for distributed operations
- Traffic and message keys
- Accumulated audit data, including records of distributed security context utilization.

Additional end system SMIB information items include:

- Key management, encipherment, integrity, and signature algorithm identifiers, and security protocol objects

- End system access control information for distributed operations
- Encryption algorithm initialization information
- Security association configuration information
- Compromise action information (e.g., revoked certificates lists)
- Contingency plan parameters (e.g., auto-purge and security policy replacement actions under emergency conditions).

Some SMIB items may be held in Directory Service Agents (DSA) for ease of access by many users. Such items might include key management information (e.g., certificates and user keying material). SMIB information stored in X.500 Directories must be integrity protected.

6.2.3 Security Protocols

Several security protocols, either existing or in development, are candidates for use in end systems implementing the DGSA. Others may be added over time.

The Transport Layer Security Protocol (TLSP) is an ISO standard (ISO, 1995b) as is the Network Layer Security Protocol (NLSP) (ISO, 1995a). The IEEE 802.10 SILS Secure Data Exchange (SDE) protocol standard (IEEE, 1992) is appropriate for LCS security services (beyond availability) when needed. MSP is the DoD standard for electronic messaging. The state of SAMP standardization was discussed in Section 5.2.8.1.

6.2.4 Cryptographic Support

The creation of distributed security contexts, which provide communications security services and strict isolation adequate for sensitive information, is usually dependent on cryptographic mechanisms. Thus, the availability of low-cost cryptographic devices is a critical element of the DGSA. These cryptographic devices must be sufficiently flexible to support requirements of different information domains in the same end system.

This flexibility will be achieved if the devices accommodate multiple cryptographic algorithms and multiple key management schemes, including public key encryption schemes and various key distribution center schemes. Otherwise, a multiplicity of cryptographic devices will be needed, resulting in increased costs. To manage these devices, there must be a registry of cryptographic algorithms and key management schemes so that the specific choices can be negotiated for a particular security association.

Currently available cryptographic and key management devices do not meet these flexibility criteria. Very large scale integration (VLSI) chip technology may now have reached a sufficient density to achieve a cost-effective single-chip design which can support multiple algorithms and a variety of key management schemes, along with a cache memory capable of handling reasonable quantities of key material. The cryptographic devices must be capable of a minimum throughput rate of 10 megabits per second to be useful with high-performance workstations. Isolation techniques must accommodate concurrent algorithm execution. In addition to creating low-cost devices, current custodial functions must be minimized through the use of electronic key management technology.

6.2.5 Distributed Management Systems

Distributed management of information systems both supports the transfer system and relies upon the transfer system for its operation. Management systems will rely upon the same transfer system security structures (distributed security contexts, security associations, and security protocols) as any other application.

When distributed information systems become very large, their management becomes very complex. To make the complexity manageable, hierarchical management approaches are often adopted. It then becomes necessary to coordinate the levels of delegated management authority. The coordination is achieved by the way management information is organized and through the control of that information as required by security policies. Hierarchical management relationships are not reflected in the way management applications communicate with one another. That is, management protocols are peer oriented, not hierarchically related. When the term "hierarchical management system" is used, it must be understood that a set of information relationships is being described, not a communications structure. This means that the hierarchical aspect of management is a human, organizational function. The organizations and administrators that manage information systems may be organized hierarchically. Management information may reflect that organization, but the end systems in which management applications are implemented only communicate as peers.

Management systems are composed of management applications implemented in end systems. Some management applications must coexist with other applications in end systems, but for logistical reasons it may be desirable to dedicate some end systems to management system activities. Management systems can be grouped into three categories based on the particular type of management function being performed. While these categories are logically separate, they often support one another. The three categories are network management, security management, and information management.

Traditional network management systems are network control centers that monitor and configure network components, perform fault isolation functions, and collect accounting and performance information. Security management systems typically provide information to support security services and mechanisms in end systems and relay systems. Most often the support is for cryptographic mechanisms, such as the DoD EKMS. Information management systems include X.500 Directory systems, the Internet Domain Name Service (DNS) and the Network Information Center (NIC).

Although these three logical categories of management systems could be implemented in end systems dedicated to the functions of only one of them, as a practical matter, some of the functions can be expected to be supported on common end systems. However, each logical category may require unique technical administrative expertise. In some cases, it will not be prudent to assign multiple administrative functions to individuals because too much control might be entrusted to them.

6.3 DGSA TRANSFER SYSTEM ISSUES

Two aspects of the DGSA transfer system deserve further discussion. One is traffic flow security, and the other is potential limitations on distributed processing functions.

6.3.1 Traffic Flow Security in Open System Communications Environments

Full TFS mechanisms are intended to conceal characteristics of communications protocols and information that might be derived from them through unimpeded observation of a communications path. Full TFS mechanisms operate at the physical protocol layer. Only if communications facilities are owned or controlled by user organizations can full TFS be applied. The use of common carrier CNs precludes the use of full TFS mechanisms. One consequence of providing full TFS between two LSEs is that the communications path cannot be used for any other purpose and, thus, creates a closed system.

The clear cost disadvantages of owning and operating private CNs means that there must be a careful examination of threats and vulnerabilities to determine whether full TFS is required. Unless it is necessary to subject all communications to full TFS, the DGSA requirements for open system and common carrier communications can be met with multiple communications connectivity. The strict isolation mechanisms required in end systems make it possible to support multiple communications connections among the information domains supported. Partial TFS mechanisms should be considered as alternatives to full TFS when judged to be appropriate to the known threats and vulnerabilities.

6.3.2 Limitations on Distributed Processing

Some communications technologies are inherently of a broadcast nature (e.g., radio, broadband LANs). Broadcast technologies make it possible to communicate with any end system that has access to the medium without the need to explicitly address information to specific end systems. Broadcast-like effects, called multicasts, can be achieved over non-broadcast communications systems through various methods that address and send information to (possibly large) groups of recipient end systems or users (e.g., groups of electronic mail recipients).

Certain limitations are encountered if cryptographic mechanisms are used to support security services for broadcast (and some multicast) communications. There are two basic choices. First, for true broadcasts, a single encryption key must be shared among all recipients. The use of a shared key among large numbers of recipients not only increases the likelihood that the key will be compromised, but the distribution and use of one or more shared keys is difficult to coordinate. (The same considerations apply to multicast services that depend on broadcast media.)

Second, for multicasts that are addressed to a group of recipients, a single key can be used for the security mechanism applied to the information to be sent and that key can be replicated and protected with a cryptographic mechanism using a different key known to each recipient.

Thus, if it is desired to broadcast information to all the members of an information domain, group multicasts are likely to be sufficient for most purposes since the member addresses are known. The only real limitation on broadcast communications is that the inherent broadcast capabilities of some media cannot be used.

This page intentionally left blank.

7.0 ADMINISTRATIVE AND ENVIRONMENTAL SECURITY

Reliance on people (i.e., administrative procedures) and the environment is an integral part of achieving total security for an information system. When products are designed and deployed in information systems, administrative and environmental conditions of their use must be met to complement the protection afforded by any hardware and software security mechanisms employed in those products. The specification of such conditions for the use of a component, facility or system is referred to as *security doctrine* in some communities. The administrative and environmental security conditions of use specify how security requirements are to be met and as such are elements of a specific security architecture. As with any design aspect of a specific security architecture, there will be different types of administrative and environmental security allocations, each with different degrees of specificity, which eventually lead to the satisfaction of the required security services through the choice of appropriate security mechanisms.[1] In the case of administrative and environmental security, security services are provided by physical, administrative, personnel, and operational security mechanisms. The DGSA suggests certain security services that can be achieved by administrative and environmental security mechanisms. The designer of more specific security architectures will need to make these, as well as more refined, choices regarding the security service allocations and types of security mechanisms. All, some, or none of the responsibility for provision of each of the security services may be allocated to administrative and environmental security mechanisms. In this section, the allocation suggestions for security services are presented and examples of administrative and environmental security mechanisms that are permissible and consistent with the DGSA are provided.

7.1 ADMINISTRATIVE AND ENVIRONMENTAL SECURITY SERVICE ALLOCATIONS AND MECHANISMS

The DGSA includes availability among the security services. In Section 3, only availability is allocated to the LCS in an LSE, while all the security services are allocated to the environment and to the end systems and relay systems. Environmental mechanisms are expected to protect the end systems, relay systems, and the LCS. Security services implemented in an LSE may take the form of physical, personnel, and administrative security mechanisms. In addition, some types of physical security mechanisms may be incorporated into the hardware of components within an LSE. The definitions of the security services of ISO 7498-2 are extended for use in the DGSA beyond only communications.

An LSE and its components must satisfy the requirements of each of the information domain security policies for which it is accredited. The administrative and environmental security mechanisms employed may vary among information domains. For example, one information domain may require authentication of the identity of an individual through cryptographic based

[1] Mechanisms, as used here, encompasses manual procedures and physical controls, as well as automated controls.

mechanisms, while another may rely on the simple possession of a badge. An LSE is the principal location for direct implementation of administrative and environmental security mechanisms, but local security mechanisms may also rely upon remote systems to provide initial capabilities and life-cycle support (e.g., key management systems, personnel investigations, shrink-wrapped software, security inspection and testing, security training and awareness).

7.1.1 Mechanisms for Identification and Authentication

Authentication of the claimed identities of individuals, as individuals or as members of a group, is a typical security policy requirement. Authentication mechanisms provide varying degrees of credibility that such claims are correct. Authentication responsibilities are often shared between administrative, environmental, and technical (i.e., hardware and software) mechanisms. Probably the most common mechanism is the picture badge and the guard. The picture on the badge matching the appearance of the holder affirms the association of the individual with what the badge represents. The identity of the individual is thereby authenticated and, in some cases, the possession of the badge establishes further claims. The reading of the magnetic code on a badge matched with the entry of a personal identification number is similar in capability to picture confirmation. Similarly, the matching of fingerprints or retina images authenticates the identity of an individual.

The use of keys with locks, passwords, or cipher lock codes authenticates identity only to the extent of the probability that the presenter is a valid holder of the object or information. That probability is based on the administrative handling and physical protection of such mechanisms or information. The same considerations apply to the use of smart cards, cryptographic ignition keys, and other credentials that make no positive connection with the holder. In general, non-forgeable information bound to the holder is the strongest type of authentication mechanism. Security mechanisms for authentication depend upon system security administrators who perform the initial assignment of the badge or other credential to an individual.

7.1.2 Mechanisms for Access Control

Access control mechanisms enforce security policy requirements for the isolation of assets and information from people and their agents. Access control mechanisms also permit authorized access to assets and information. The first line of protection for the LSE is through mechanisms that control access to the facilities (e.g., buildings, rooms) containing the end systems, relay systems, and LCSs. The human security guard is one of the most familiar types of access control mechanisms. Key, combination, and cypher locks are common mechanisms for controlling access to facilities. Placing an entire LSE within a vault is an extreme form of facility control. With the assumption that only authorized people are in the LSE, surveillance of their activities by security administrators or by co-workers can form the next line of protection. Areas may be declared to require at least two people to be present when activities are in progress ("no-lone" zones).

The next line of protection involves the use of approved containers (e.g., combination safes and locking cabinets) for the protection of system assets. Such containers can be used to protect entire system components (end systems, relay systems, and LCSs) or information storage media (e.g., disks, tapes). Finally, the components themselves may contain access control mechanisms such as power locks, two-person-control devices, and sealed housings.

Within and beyond these lines of protection, access control becomes the responsibility of hardware and software features of the end systems and relay systems. Access control mechanisms can also contribute to the provision of confidentiality, integrity, and availability services; independent aspects of these services are presented in the following sections.

7.1.3 Mechanisms for Confidentiality

Confidentiality mechanisms satisfy security policy requirements to protect information from unauthorized disclosure. The major applications of administrative and environmental confidentiality mechanisms in LSEs involve video displays, printing devices, sounds, and non-video electromagnetic emanations.

Users and security administrators can control when, where, and in whose presence video information is displayed. Video display emanations can be controlled through screen filters and shielded enclosures. Printer ribbon handling, copy counting, and labeling requirements can be controlled by users, operators, and system administrators. The control of trash and the destruction of paper and other media are important procedures. Paper shredders may be useful. Procedures for handling and mechanisms for erasure of persistent storage media can be critical to confidentiality. Sound insulation and sound masking can be used to control disclosure through conversations and machine noises. Electromagnetic emanations, either radiated or conducted, can be confined by shielding rooms and by filtering signal and power wiring using standard TEMPEST features. The presence of copiers and photographic equipment in LSEs requires careful control. Paper and other media devices should be properly wrapped prior to shipping or mailing.

7.1.4 Mechanisms for Integrity

Integrity mechanisms are used in response to security policy requirements to protect information and other system assets from unauthorized modification. The major applications of administrative and environmental integrity mechanisms in LSEs involve the correctness of end system and relay system hardware and software, and the correct functioning and use of other administrative and environmental security mechanisms. System components may have features that permit security diagnostic checking of hardware (for example, through comparison of diagnostic known-answer tests with off-line security check mechanisms). Non-forgeable seals and protective coatings may be used on hardware components and subcomponents to detect or prevent alteration. Cryptographic and non-cryptographic check value mechanisms can be used to ensure the integrity of software packages as delivered and as used.

Regular inspections of facilities and system components is an important part of using integrity mechanisms. Devices used for integrity checking must be stored in protected areas. Software master copies and small system components must also be stored in protected areas while not in use. Protection from electromagnetic interference can be accomplished by filtering and shielding.

7.1.5 Mechanisms for Non-Repudiation

Non-repudiation mechanisms support security policy requirements for proof of delivery and proof of origin of information transactions. Non-repudiation mechanisms may include the contents of a transaction. For paper transactions, notary services and personal signatures are useful mechanisms in providing non-repudiation services. Non-repudiation mechanisms, such as hash coding of data and digital signatures, can be used to validate the source of software packages. Non-repudiation mechanisms could be used for verifying that hardware is unchanged from its manufactured state.

7.1.6 Mechanisms for Availability

Availability mechanisms in communications networks and LSEs satisfy security policy requirements for availability of communications and processing resources. The ability of communications networks to provide timely and regular service depends upon the total security architecture, implementation, and management of those systems. The techniques of redundancy, diversity, contingency reserves, and contingency planning play a large part in communications network availability. Within LSEs, the LCS must be similarly designed and protected to avoid failure outages. Generally, the physical protection and integrity checking of the end systems, relay systems, and LCSs will provide for their availability.

7.2 COTS PRODUCT CONSIDERATIONS

Current COTS products may lack built-in security mechanisms such as those presented in the previous section. Therefore, additional procedures may be required or separate COTS tools that provide a measure of security assurance. COTS products may also be vulnerable to component modification and substitution. Any user not being closely observed may be able to modify or substitute COTS product components to their own benefit or the detriment of the organization. The administrative and environmental mechanisms must ensure that COTS products can be physically accessed only by persons authorized for access to all information in the component unless escorted by someone who is so authorized. At the other extreme, when sufficient built-in isolation mechanisms exist (in GOTS products or custom-designed products), then all communities of interest can be satisfied that physical access is permissible by persons authorized in only one information domain of all those supported.

7.3 SECURITY MANAGEMENT

Security management, as presented in Section 5, includes security service management, security mechanism management, and the management of all security aspects of the system. All of these functions are performed within an LSE. The information domain security manager is an administrator who is authorized to perform installation and maintenance of the information domain security policy representation, access control lists, and other items of the SMIB, such as cryptographic keys. The security manager is provided tools, such as a SMAP, to perform these tasks. The security manager is ultimately responsible for checking personnel clearances, monitoring guard activities, performing audits of security-relevant records, and, in general, supporting all other security mechanisms.

The security aspects of system management are no different from any other applications which require protection. The system must have a security policy and administrative and environmental security mechanisms will be used in support of system management activities. A critical aspect of security management is the training of security administrators and users so that they understand their responsibilities as part of the entire security posture.

This page intentionally left blank.

8.0 EXAMPLE OF A HIGH-LEVEL ARCHITECTURE BASED ON THE DGSA

This section presents an example of how the DGSA's concepts work together and how the DGSA could be used in a typical networked environment. This example is based on a Group Medical Practice (GMP). A GMP was selected to provide an example with which most readers would be familiar and a sufficiently rich environment to demonstrate the concepts of the DGSA. A more detailed example, included in the DGSA Version 1.0, is separately available in "Detailed DGSA Example: Drug Enforcement." Note to demonstrate the concepts of the DGSA, specific detail is provided where necessary. In an actual GMP, additional functions and types of information would be used and additional relationships would exist with internal and external organizations. A number of assumptions are made in this example to facilitate the demonstration of DGSA concepts. These assumptions do not necessarily reflect the operation of an actual GMP, and the reader is cautioned that certain assumptions may invalidate the example in specific legal jurisdictions.

8.1 MISSION

The first stage in developing an information system security architecture is to understand the missions of the organization using the information system. As discussed in Section 2, every organization has missions or goals. For this example, the mission of the GMP is to provide quality health care at a reasonable cost. Most organizations are divided into components, each with its own mission that support the overall mission. Some components of the GMP are the care providers, business office, and laboratories. The care provider's mission is to treat patients according to the principles of the medical profession. The business office's mission is to manage the financial activities of the GMP. The laboratories' mission is to perform medical tests accurately.

8.2 POLICY

Once the GMP mission is determined, the organization must develop a security policy for that mission. The security policy should include requirements from a variety of sources, such as laws and corporate directives. For the GMP example, federal and state laws on privacy require the protection of patient information including the patient's medical, financial, and personal information. Corporate directives define methods of protecting the personnel data on the GMP's care providers and laboratory workers. For example, the GMP's security policy states that only the personnel department, the supervisor, and the employee may access an employee's personnel folder.

Another source of requirements for the GMP security policy is the perceived threat environment. Threats can be internal or external. An example of an internal threat is the embezzlement of GMP funds. An example of an external threat is a tabloid attempting to access a patient's medical history. For the GMP example, the threats are primarily aimed at the integrity and confidentiality of the GMP's information objects. This threat environment leads to requirements for high strength of service for identification and authentication (I&A), confidentiality, and integrity. These requirements, combined with the requirements derived from the laws and corporate directives, generate the GMP security policy. The GMP security policy serves as the common basis for the development of security policies for each of the information domains.

8.3 INFORMATION DOMAINS

An information domain as defined in Section 3 is a set of users, their information objects, and a security policy. The security policy for the GMP identifies information domains and their constituent elements. These information domains are tied directly to the missions that they support. Some of the GMP's information domains are the patient medical history, patient financial information, laboratory records, accounting, and patient address information. Each of these information domains supports one or more of the GMP's missions. For example, the patient address information domain is constructed to support the care providing and business office missions. While the GMP example uses a number of information domains, only the patient medical history information domain is presented in depth. A patient medical history information domain is created for each patient in the GMP. For this example, it is assumed that all patients have a primary medical care provider or doctor.

The set of users of the patient medical history information domain includes the patient, the patient's doctors, their nurses, and the medical director of the GMP. This membership limits the access to a patient's medical history to only those individuals directly involved with the patient. The medical director has access for emergency situations and for internal situations in the GMP. Membership in the information domain is not static. Staff turnover or the need for consultation by a specialist will cause changes in the membership of the information domain. The patient's primary doctor has the authority to modify the membership of this information domain.

Examples of information objects within the patient medical history information domain include test results, prescriptions, and reports on a patient's medical visits. Each of these information objects is uniquely identifiable and directly associated with its information domain. In addition, the GMP requires protection of each information object in the patient medical history information domain to ensure the integrity and authenticity of the data.

The final, and perhaps the most critical, element of the information domain is the information domain security policy. The information domain security policy comprises the roles and privileges of the members and the protections that must be applied to the information objects within the information domain and the transfer policy. The transfer policy addresses inter-domain and intra-domain transfers of the information objects. The information domain security policy identifies the security services required for operation within an information domain. Each security service has a strength of service characteristic. For the GMP example, the value of the strength of service is specified as a low, medium or high level of assurance.

The roles, privileges, and protections of the patient medical history information domain security policy are:

- Membership in the patient medical history information domain includes the patients, their doctors and nurses, and the medical director.

- All members of the information domain must be identified and authenticated at a high level of assurance.

- Every member is allowed to view the information objects in the information domain.

- Members cannot modify the contents of any of the existing information objects.

- The deletion of information objects in the information domain requires the consent of both the patient and the doctor of record.

- The integrity of the information objects in the information domain must be protected at a high level of assurance.

- The confidentiality of the information objects in the information domain must be protected at a high level of assurance.

- The identity of the creator of an information object must be protected. Therefore, non-repudiation of origin of an information object in this information domain must be protected at a high level of assurance.

- The availability of the information objects of the patient medical history information domain is at a moderate level of assurance.

The transfer policy for the patient medical history information domain is:

- The confidentiality of all information objects must be maintained during inter- and intra-domain transfers at a high level of assurance.

- All outgoing inter-domain transfers of information objects must be approved by the patient before the data can be transferred to another information domain.

- In an emergency, such as patient incapacitation, the medical director is authorized to release the patient's medical information to a physician treating the patient who is not already a member of the information domain.

- All incoming inter-domain transfers of information objects are accepted, if the integrity of the information objects is verifiable and if they pertain to the patient.

The described patient medical history information domain security policy may not address every issue of a patient's medical records, but serves as an example of the type of material in an information domain security policy. This material is derived from the mission requirements and the GMP security policy. It should be noted that, while there must be a patient medical history information domain for every GMP patient, the same information domain security policy can be used. If exceptional circumstances arise, the basic information domain security policy can be modified.

Section 8.5 presents three scenarios to demonstrate the concept of operations for a DGSA based architecture. These scenarios require a variety of information domains to demonstrate the concepts of the DGSA. Figure 8-1 lists the information domains used in each scenario.

8.4 INFORMATION SYSTEM SECURITY ARCHITECTURE

This section presents a GMP information system security architecture based on the DGSA concepts. In this system, end systems are available to all employees of the GMP (e.g., doctors, nurses, or administrative staff), all patients, and all organizations (e.g., hospitals, insurance firms). Access rights to the respective information domains vary depending on the role of the GMP employee or the association with the patient (e.g., the patient's primary physician, hospital nurse).

New Patient Enrollment	Medical Visit	Hospital Admission
Patient Address Information(address, phone) Domain	Patient Address Information Domain	Patient Address Information Domain
Patient Financial Information Domain	Child Medical History Information Domain	Hospital Billing and Insurance Information Domain
Patient Medical History Information Domain	Patient Medical History Information Domain	Patient Medical History Information Domain
Doctor Appointment Calendar Information Domain	Lab Appointment Calendar Information Domain	Hospital Patient Medical History Information Domain
Accounting Information Domain	GMP Lab Information Domain	Hospital Lab Information Domain
Security Management Information Domain	Security Management Information Domain	Security Management Information Domain

Figure 8-1. Information Domains for the Scenarios

Figure 8-2 depicts an example GMP architecture. Although a GMP may have other end systems for additional components, only the four end systems shown below are specifically addressed in the example scenarios. For each of the four end systems discussed, the end system security policy, functionality, and security service allocations are identified based on the information domains that are supported by those end systems. For an information domain to be supported on an end system, the end system must be capable of implementing the information domain security policy. For this example, a single security management information domain is maintained across all end systems and information domains of the GMP in accordance with the one to many paradigm described in Section 5. The security management information domain contains the information domain security policies and other security critical information objects.

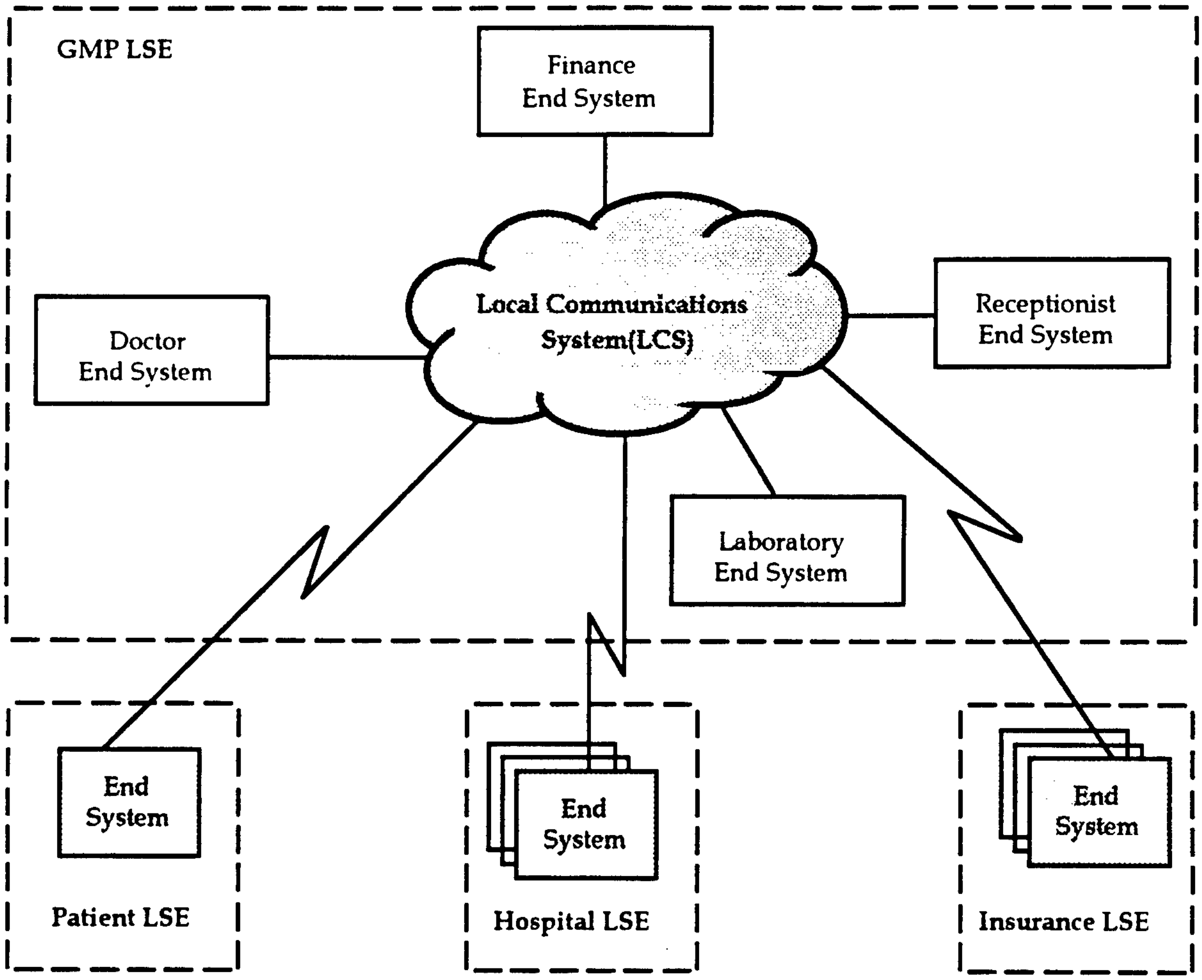


Figure 8-2. Architecture for GMP Example

The receptionist end system is used to schedule and check appointments and maintain calendars for the doctors of the GMP. This system is used to maintain records of patient identification information, such as address and telephone number. The receptionist end system provides support for the patient address information domain and the appointment calendar information domain. The receptionist end system must provide strict isolation at a moderate level of assurance to support the information domains that are resident on this end system. The receptionist end system must provide confidentiality and I&A security services at a medium level of assurance to ensure that patient address and doctors' calendar information is not released outside the membership of the respective information domains. A low level of assurance is required for the security services of integrity, non-repudiation, and availability.

The laboratory staff uses the lab end system to schedule tests, to monitor lab personnel availability, and to record test results. The lab end system provides support for the medical history information domain, patient address information domain, lab results information domain, lab calendar information domain, and GMP security management information domain. The lab end system must provide strict isolation at a high level of assurance to support the information domains that are resident on this end system. The lab end system must provide confidentiality, integrity, I&A, and non-repudiation security services at a high level of assurance. These services ensure that medical history and lab result information is not released outside the membership of the respective information domains, patient privacy is protected, and medical history and test results cannot be altered. The requirement for availability security services is a medium level of assurance.

The doctor end system is used by a doctor of the GMP to record and update patient medical history records. Although a patient may request a copy of the records for their own end system, the doctor's version of these records is the master copy. The doctor end system provides support for the medical history information domain, patient address information domain, doctors' calendar information domain, and GMP security management information domain. This end system must provide strict isolation at a high level of assurance to support the information domains that are resident on this end system. This end system must provide confidentiality, integrity, I&A, and non-repudiation security services at a high level of assurance. These services ensure that medical history information is not released outside the membership of that information domain, patient privacy is protected, and patient medical histories cannot be altered. The requirement for availability security services is a medium level of assurance.

The finance end system is used by the financial staff of the GMP to record patient insurance information, patient billing information, and insurance billing and payments. The finance end system provides support for the patient address information domain, patient financial information domain, accounting information domain, and the GMP security management information domain. This end system must provide strict isolation at a moderate level of assurance to support the information domains that are resident on this end system. This end system must provide integrity and identification and authentication security services at a medium level of assurance. These services ensure that the GMP's financial information and billing information is not released outside the membership of the finance information domain and cannot be altered or deleted. The policy requires confidentiality and non-repudiation security services at a medium level of assurance to ensure that adequate protection of patient financial information. The requirement for availability security services is low, since the system need only be available when the financial office is open (e.g., Monday through Friday, 8:30 AM to 5:30 PM).

The security policy for the security management information domain of the GMP end systems indicates that security mechanisms must be available on all end systems to support the establishment of information domains. For example, mechanisms are required to create the memberships and to install the security policies of the various information domains.

In addition to the security service allocations and strict isolation requirements identified above, the overall GMP LSE security policy requires that the LCS provide availability security services at a medium level of assurance to ensure that all GMP end systems are able to communicate as needed when the GMP is open. The LCS is also required to provide confidentiality and integrity security services to protect the information in transmission within the GMP at a medium level of assurance. The GMP LSE security policy requires that administrative and environmental controls at a medium level of assurance be in place to safeguard physical access to all GMP end systems.

Figure 8-3 provides a mapping between the requirements identified for the medical history information domain and the security service allocations across the end systems of the GMP. All requirements of the medical history information domain are addressed by the allocation of security services to the doctor and lab end systems. On the bases of the allocations of security services, security mechanisms can be chosen to provide the requisite strength of service.

The Hospital LSE is composed of a collection of end systems that serve different purposes (e.g., financial, patient check-in). This collection of end systems is treated as a single end system here to simplify this example. The hospital end system communicates with the GMP doctor end system. The Hospital Patient Medical History information domain is created on the GMP doctor end system.

The Insurance LSE, like the Hospital LSE, is actually a collection of end systems that serve different purposes (e.g., billing receipt, requests for insurance information, insurance claims, payments made). This collection is treated as a single insurance end system in this example for simplicity. The insurance end system communicates with the GMP finance end system.

The patient end system may be used to communicate with:

- The receptionist end system of the GMP to establish appointments

- The hospital end system to establish hospital test or lab appointments

- The insurance end system to identify any errors or to present the issues of a specific case.

Each patient is assumed to have an end system readily available to them.

Medical History Information Domain Requirements	Security Service Allocation Within the GMP
Each information object is digitally signed to prevent modification	High assurance integrity security services for the doctor and lab end systems
Membership limited to patient, doctor, nurses, and medical director and strong I&A is applied	High assurance identification and authentication security services for the doctor and lab end systems
Objects only deleted by joint permission of patient and doctor	High assurance integrity security services for the doctor and lab end systems
Integrity of patient medical information must be maintained	High assurance integrity security services for the doctor and lab end systems
Confidentiality of patient medical information must be maintained	High assurance confidentiality security services for the doctor and lab end systems
Information object creator must be identifiable	High assurance non-repudiation security services for the doctor and lab end systems
Patient medical histories must be reasonably available	Medium assurance availability security services for the LCS and ESs
Confidentiality must be maintained during any information transfers	High assurance confidentiality security services for the doctor and lab end systems
Outbound inter-domain transfers must be approved by the patient	High assurance access control security services for the doctor and lab end systems
Medical director can release medical information in an emergency	High assurance access control security services for the doctor and lab end systems
Incoming inter-domain transfers must be verifiable and pertinent to the patient	High assurance integrity and non-repudiation security services for the doctor and lab end systems

Figure 8-3. Mapping of Requirements to Security Service Allocations

8.5 SCENARIOS

This section presents three scenarios for the GMP example. These scenarios demonstrate:

- The creation and instantiation of information domains
- Creation of information objects
- Creation and use of security contexts and security associations
- Use and establishment of access privileges
- Transfer of information objects between information domains (inter-domain and intra-domain)
- Creation and use of multidomain objects
- Switching from one information domain to another.

For all three scenarios, the doctor and patient jointly control the information in the patient's medical history information domain. Doctors may create new information objects in this information domain and read any existing information objects in this information domain. When an information object is created, it must be signed (using a digital signature that is public key based) to protect its integrity. That is, once a medical history information object has been created for a patient it must not be altered. Patient medical history information objects must not be deleted without the consent of both the doctor and the patient. A patient may obtain a copy of any of his medical history information objects. The copy of the medical history information object retains the digital signature of the originator and therefore cannot be modified without being detected.

A doctor can transfer copies (e.g., transfer specific records) to different information domains, for example to a hospital information domain, with the consent of the patient. Medical specialists can become members of the information domain on a temporary basis if the doctor and patient both agree to permit the specialist to access the patient information. Alternatively, a temporary information domain containing copies of only the pertinent medical history information can be created with the doctor, the patient, and the specialist as its members.

Normally, a hospital obtains a copy of selected information objects, as necessary, from the doctor via an information transfer. The hospital has one information domain per patient and appropriate hospital employees have read access to all information in that information domain. Appropriate hospital staff may also add new information objects to the information domain, as necessary. A transfer policy permits information to be sent between a hospital information domain and the patient's medical history information domain. When new objects are introduced into the hospital information domain, a copy of the object is immediately transferred into the patient's medical history information domain. All objects in the hospital information domain carry a digital signature and cannot be modified by the patient.

Patient information is dispersed among several information domains. Multidomain objects are created to simplify the presentation of patient information. For example, the finance information domain contains all patient billing and insurance related information objects, and the personal address information domain contains all identification information for the patient. A multidomain object is created which links these information objects to facilitate the operation of the finance mission. These information objects provide a look-at-a-glance for all GMP financial staff regarding patient information. Similar links are created among other information domains in the GMP but are not described further in this example.

8.5.1 Scenario 1: New Patient Enrollment

A request for an appointment is sent from the patient's end system to the receptionist end system at the GMP. The receptionist schedules an appointment based on the patient's name and sends an acknowledging message to the patient's end system.

Upon arrival at the GMP for the first appointment, the receptionist creates an information object for the patient in the GMP patient address information domain. The information object contains identification information, such as patient name, address, home telephone number, and work telephone number. The identification information is then transferred from the receptionist end system to the finance end system based on the intra-domain transfer policy for the patient address information domain.

The new patient then speaks with a financial staff representative who obtains additional information, regarding responsibility for bills and insurance coverage. A financial information object is created in the finance information domain using that information. The financial staff representative establishes a multidomain object for the patient in the accounting information domain. That information object points to the patient address information object in the patient address information domain and the finance information object in the financial information domain.

The doctor's nurse creates a medical history information domain for the new patient and obtains a medical history from the patient. The medical history is recorded as information objects in the new medical history information domain. From this point forward, access to this new medical history information domain requires user authentication.

At the first doctor/patient meeting the patient's medical history is reviewed. The doctor may update the patient's medical history information objects after their initial consultation. After reviewing the patient's history, the doctor uses a digital signature to sign the information objects so that they may no longer be altered. The doctor creates new information objects in the medical history information domain to record the events of this appointment. If the doctor requires a follow-up appointment or lab tests, a message is sent to the receptionist requesting that such appointments be scheduled before the patient leaves.

If necessary, the GMP receptionist updates the doctor's appointment calendar by establishing another appointment with the patient. The receptionist end system transfers a copy of the patient address information for this patient to the patient address information domain on the lab end system and requests test scheduling. The lab end system creates a new object for this patient in the lab test information domain and establishes an appointment. Later, a multidomain information object is created that contains pointers to the patient identification information and the patient test results information.

8.5.2 Scenario 2: Medical Visit

This scenario builds upon the new patient enrollment scenario. The patient makes an appointment with his or her doctor through the receptionist, as described in the previous scenario. When the patient visits the doctor, the doctor first authenticates himself to the end system in order to accesses the patient's medical information to review the patient's status.

The patient is suffering from a minor ailment, but as a preventive measure, the doctor orders lab tests. Until the laboratory results have been completed, the doctor issues an interim prescription to alleviate the patient's ailment. The doctor creates the prescription on his end system and sends a copy of the prescription electronically to the patient's pharmacist. The doctor digitally signs the prescription, so that the pharmacist can verify its integrity and authenticity. The prescription is encrypted, in accordance with the patient's privacy requirements, during transmission through the network. This transfer is accomplished by accessing the pharmacist's certificate stored in the public key certificate directory and using the pharmacist's public key for encryption.

During this visit, the patient asks a question about the results of tests done for the patient's child. It is assumed for this scenario that the patient is the child's legal guardian and that the doctor is the primary care provider for the child. The doctor attempts to access the child's medical information. Since the end system has previously authenticated the doctor, the end system must only determine whether the doctor is a member of the child's medical history information domain. After verifying that the doctor is a member of that information domain, the system grants access to the data and the doctor is able to answer the parent's question. This scenario assumes that the child's information is available on this end system. If the information is stored on a different end system then the doctor's end system must make a connection to another end system in the GMP, such as a database server. The doctor then accesses the information directly on the server or the information object is transferred to the doctor's end system. In any case, the end system must ensure that the confidentiality of the information is protected while it moves through the GMP local communication system. There is a high assurance requirement for the confidentiality of this information. Note this requirement may have been satisfied through extensive environmental and administrative procedures used by the GMP to protect its local subscriber environment and a simple cryptographic mechanism.

After the patient's visit has ended, the doctor completes a report for the visit which becomes a new information object within the patient's medical history information domain after the doctor has digitally signed it. The doctor sends a statement to the financial office so that the patient's insurance company can be billed for the routine medical visit. Upon receipt of the doctor's statement, the financial office creates a bill within the financial information domain. Since the

GMP previously established a transfer policy with the insurance company, the bill is transferred to the insurance company in accordance with this policy.

8.5.3 Scenario 3: Hospital Admission

The results of the laboratory tests conducted during the patient's visit indicate a more serious medical problem that requires a short stay in the hospital. The doctor creates a tentative transfer request for the patient medical information that is needed by the hospital. The doctor arranges for the patient to return for an office visit to discuss the results of the laboratory tests. If the patient agrees on the need for the hospital stay, the patient must give electronic consent before the doctor's end system releases the patient's medical information to the hospital. After the patient is authenticated by the doctor's end system, the patient reviews the transfer request, and, assuming concurrence, the patient digitally signs the transfer request.

The doctor, as a member of the staff of the hospital, makes arrangements for the patient to enter the hospital. The hospital creates a medical history information domain for the patient on the hospital end system. The doctor then initiates the transfer of the patient's medical information. The security policy enforcement function on the doctor's end system checks to see that both the doctor and the patient consented to the transfer before releasing the data to the hospital's patient medical history information domain. (In an emergency, this information could be released by the medical director of the GMP without the patient's consent.)

The actual transfer is accomplished by creating the patient's hospital medical history information domain on the doctor's end system (in accordance with prior agreements between the hospital and the GMP). An application on the doctor's end system causes the creation of security contexts for both the GMP and hospital patient medical record information domains and the previously approved interdomain transfer takes place. The transfer of information to the hospital end system requires the establishment of a security association between the doctor's end system and the hospital's end system over their common communications network. The security association maintains the confidentiality of the information during transfer. The first step in creating the security association is for the doctor's end system to verify that it is connected to the hospital's end system. Once this connection has been confirmed, the end systems security management functions negotiate the parameters of the security association to satisfy the requirements of the hospital patient medical record information domain's transfer policy. Since the communication network only provides the security service of availability, a strong cryptographic mechanism is employed to provide the requisite level of confidentiality. The completion of the negotiation establishes a distributed security context between the two end systems and the secure transfer of the information objects.

During the patient's stay in the hospital, any medical information objects that are created by the hospital are transferred into the GMP's patient medical history information domain. The process is the reverse of that used for the transfer into the hospital's medical history information domain. After the completion of the patient's stay in the hospital, the hospital archives the patient medical history information domain.

APPENDIX A

REFERENCES ;

Note: References appearing in this section represent documents used in preparation of this volume, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

1. Abrams, Marshall D. and Michael V. Joyce, January 1993, *On TCB Subsets and Trusted Object Management*, MITRE Technical Report 92W0000248, McLean, VA.

2. Center for Information Systems Security (CISS), 30 January 1995, *Department of Defense Goal Security Architecture Transition Plan,* Defense Information Systems Agency, Washington, DC.

3. Case, Jeff, Mark Fedor, Martin Schoffstall and James Davin, 1989, *Simple Network Management Protocol (SNMP)*, Internet Request for Comments 1098.

4. Case, Jeff, 1991, *SNMP Version 2*, Internet Request for Comments 1441.

5. International Telegraph and Telephone Consultative Committee (CCITT), 1988, *Recommendations X.400-X.420: Data Communications Networks, Message Handling Systems.*

6. _____________, 1992, *Recommendations X.500-X.521 Data Communications Networks, Directory.*

7. Department of Defense, 1985, *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, Washington, DC.

8. Institute of Electrical and Electronic Engineers (IEEE), *Standard for Interoperable LAN/MAN Security, Part 2--Secure Data Exchange Protocol Specification*, IEEE 802.10b.

9. _____________, 1995, IEEE *Standard for Interoperable LAN/MAN Security, Clause 3-Key Management Protocol Specification (Draft)*, IEEE 802.10c.

10. International Organization for Standardization (ISO), 1989a, *Information Processing Systems -- Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture*, ISO 7498-2.

11. _____________, 1989b, *Information Processing Systems -- Open Systems Interconnection -- Basic Reference Model -- Part 4: Management Framework*, ISO 7498-4.

12. _____________, 1991, *Information Technology -- Common Management Information Protocol -- Part 1: Specification*, ISO/IEC 9596-1.

13. ______________, 1994a, *Information Technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*, ISO/IEC 7498-1.

14. ______________, 1994b, *Information Technology -- Open Systems Interconnection -- General Upper Layers Security -- Part 3: Security Exchange Service Element (SESE) Protocol Specification*, ISO/IEC 11586-3.

15. ______________, 1995a, *Information Technology -- Open Systems Interconnection -- Network Layer Security Protocol*, ISO/IEC 11577.

16. ______________, 1995b, *Information Technology -- Telecommunications and Information Exchange Between Systems -- Transport Layer Security Protocol*, ISO/IEC 10736.

17. ______________, 1995c, Information Technology -- *Open Systems Interconnection - Security Frameworks for Open Systems - Part 1: Overview*, ISO/IEC DIS 10181-1.

18. ______________, 1995d, Information Technology -- *Open Systems Interconnection - Security Frameworks for Open Systems - Part 3: Access Control*, ISO/IEC 10181-3.

19. Linn, John, September, 1993, *General Security Services Application Program Interface*, Internet Engineering Task Force, Request for Comments: 1508.

20. National Security Agency, 22 February 1993, Draft *Department of Defense Information Systems Security Policy*, DISSP-SP.1.

21. Rushby, J., September, 1984, A Trusted Computing Base for Embedded Systems, Proceedings of the 7th DOD/NBS Computer Security Symposium, pp. 294-311.

APPENDIX B

ACRONYMS

ADF	Access Control Decision Function
AEF	Access Control Enforcement Function
API	Application Program Interface
C&A	Certification & Accreditation
C4I	Command, Control, Communications, Computers, and Intelligence
C4IFTW	C4I for the Warrior
CCITT	International Telegraph and Telephone Consultative Committee
CIM	Center for Information Management
CISS	Center for Information System Security
CMIP	Common Management Information Protocol
CN	Communications Network
COTS	Commercial-Off-the-Shelf
DGSA	DoD Goal Security Architecture
DIS	Defense Information System
DISA	Defense Information Systems Agency
DISN	Defense Information System Network
DISSP	Defense Information Systems Security Program
DMS	Defense Message System
DNS	Domain Name Service
DoD	Department of Defense
DSA	Directory Service Agents
EKMS	Electronic Key Management System
ES	End System
GMP	Group Medical Practice
GOTS	Government-Off-the-Shelf
GSS	General Security Service
GULS	General Upper Layer Security
I&A	Identification and Authentication
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
ITSDN	Integrated Tactical/Strategic Data Network
LAN	Local Area Network
LCS	Local Communications System

LMD	Local Management Device
LSE	Local Subscriber Environment
MAP	Management Application Process
MAN	Metropolitan Area Network
MIB	Management Information Base
MISSI	Multilevel Information System Security Initiative
MLS	Multilevel Security
MSP	Message Security Protocol
NIC	Network Information Center
NLSP	Network Layer Security Protocol
NSA	National Security Agency
OSD	Office of the Secretary of Defense
OSI	Open Systems Interconnection
RM	Reference Model
RS	Relay System
RVM	Reference Validation Mechanism
SAMP	Security Association Management Protocol
SDE	Secure Data Exchange
SDNS	Secure Data Network System
SESEP	Security Exchange Service Element Protocol
SILS	Secure Interoperable LAN/MAN Standard
SMAP	Security Management Application Process
SMIB	Security Management Information Base
SNMP	Simple Network Management Protocol
SPDF	Security Policy Decision Function
SPEF	Security Policy Enforcement Function
TAFIM	Technical Architecture Framework for Information Management
TFS	Traffic Flow Security
TLSP	Trasport Layer Security Protocol
U.S.	United States
VLSI	Very Large Scale Integration
WWW	World Wide Web

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 7: Adopted Information Technology Standards (AITS)

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents

- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.

- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES

This version of Volume 7 of the TAFIM was preceded by a version labeled as Version 3.0 DRAFT, dated 30 September 1995.

In addition, an interim version of this volume, labeled Version 2.1, was also released in September 1995. The comment resolution period for this interim version was still ongoing at the time the official TAFIM draft was prepared; as a consequence, the standards in this draft are still subject to change based on issues remaining unresolved at this point.

As part of the process of harmonizing Volumes 2 and 7 of the TAFIM, Internationalization Services has been added to Volume 7. Some of the standards in this area are currently under study. Other major additions to this version are the result of the harmonization of taxonomy and terms between Volumes 2 and 7.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.
- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.
- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.
- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.
- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. DISA is investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1: Overview	3.0	30 April 1996
Volume 2: Technical Reference Model	3.0	30 April 1996
Volume 3: Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4: DoD SBA Planning Guide	3.0	30 April 1996
Volume 5: Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6: DoD Goal Security Architecture	3.0	30 April 1996
Volume 7: Adopted Information Technology Standards	3.0	30 April 1996
Volume 8: HCI Style Guide	3.0	30 April 1996

Other working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

PREFACE

The Adopted Information Technology Standards (AITS) is a product of the Department of Defense (DoD) Defense Information Systems Agency (DISA), Joint Interoperability and Engineering Organization (JIEO), Center for Standards (CFS). It was developed with support from the DoD Commanders-in-Chief (CINCs), Services, and Agencies and was approved by the Standards Coordinating Committee (SCC). Further standards guidance can be found in the AITS companion document, the Information Technology Standards Guidance (ITSG). Both documents may be obtained from:

Help Desk

Department of Defense

Standards Assistance Directorate

DISA/JIEO/CFS/JEBD

10701 Parkridge Blvd

Reston, Virginia 22091-4398

e-mail: Helpdesk@jcdbs.2000.disa.mil

This page intentionally left blank.

CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

1.1 PURPOSE

The purpose of the AITS is to guide DoD Enterprise acquisitions and the migration of legacy systems by providing a definitive set of information technology (IT) standards to be used in DoD. These standards provide consistency across the Enterprise, Mission, Function, and Application levels of the DoD Integration Model, as described in Volume 1 of the TAFIM. The goal in providing effective and usable standards guidance is to support the broader TAFIM objectives of:

- Improving user productivity
- Improving development efficiency
- Improving portability and scalability
- Improving interoperability
- Promoting vendor independence
- Reducing life cycle costs
- Improving security
- Improving manageability.

1.2 SCOPE

The AITS is the definitive set of IT standards to be used in the DoD. The AITS applies to all DoD IT programs and initiatives. The AITS is the common DoD IT standards reference applicable to all life-cycle decisions affecting interoperability, portability, and scalability, and is to be used to guide in the development of standards profiles. The Information Technology Standards Guidance (ITSG) provides a foundation for the AITS. The ITSG contains more detailed, supporting information about the state of standardization in each of the subject areas listed in the AITS Figures (see Appendix A), as well as other areas in which standardization has not progressed to the point where adopting a standard is in order (see Section 1.3). These subject areas are called base service areas (BSAs).

The term *adopted* is used to mean that the standards and specifications in the AITS are approved by DoD for use in satisfying each BSA function. This standards guidance is applicable to all systems and programs whether at the leading edge of technology or preserving current operational capability in a long-standing legacy system. Migration toward open system environments (OSE) remains an ever-present goal, because of the enhancement of competition, interoperability, and portability. The following recent directives and instructions were published to support the goal:

- DoD Directive (DoDD) 4630.5, *Compatibility and Interoperability of Tactical Command, Control, Communications, and Intelligence Systems*, promulgated in November 1992, requires that procedures be established for the development, coordination, review, and validation of compatibility, interoperability, and integration of Command, Control, Communications, and Intelligence (C3I) systems. It further stipulates that *all* C3I systems developed for use by U.S. forces are considered to be for joint use.

- DoD Instruction (DoDI) 4630.8, also promulgated in November 1992, directs that the Chairman of the Joint Chiefs of Staff (CJCS) provide amplifying instructions for implementing DoDD 4630.5. DoDI 4630.8 also stipulates that the CFS is responsible for evaluating program acquisition documents (Mission Need Statements (MNSs), Operational Requirements Documents (ORDs), and Test and Evaluation Master Plans (TEMPs)) from an IT standards perspective and that an IT standards profile be developed and submitted for CFS review no later than Milestone II.

- In January 1993, DoDI 8120.2, *Automated Information System (AIS) Life Cycle Management (LCM) Process, Review, and Milestone Approval Procedures*, was promulgated, stipulating that all automated information systems (AISs) programs incorporate standards planning, including the development of IT standards profiles per the TAFIM.

- The entire policy came together in July 1993, with the promulgation of CJCS Instruction (CJCSI) 6212.01, implementing DoDD 4630.5 per direction by DoDI 4630.8. CJCSI 6212.01, replacing MOP 160, effectively combined policies stipulated by DoDD 4630.5 and DoDI 8120.2 by expanding the scope of the CJCS's responsibility for the development, coordination, review, and validation of compatibility, interoperability, and integration of Command, Control, Communications, Computers, and Intelligence (C4I) systems. The fourth *C* (computers) was intended to account for AIS (primarily business systems) under DoDI 8120.2.

At this point, DoD policy clearly stipulates that all C4I systems, now covering the entire spectrum of the DoD Enterprise Model, are required to produce IT standards profiles requiring certification by the CFS.

The AITS does not contain data administration policy, standards, or procedures. These can be found in DoDD 8320.1, *Data Administration*, September 26, 1991, and DoD 8320.1-M-1, *DoD Data Element Standardization*, March 1994.

The adopted standards in the AITS are derived from a larger volume, the *Information Technology Standards Guidance* (ITSG). The AITS and ITSG work together, but perform very different roles. The AITS is intended to contain summary information (i.e., What are the adopted standards?). The role of the ITSG is to provide additional, supporting details about the standards in the AITS, including other related standards and emerging standards (i.e., What else might I need to know other than the fact this standard is adopted?). Figure 1-1 depicts the relationship of AITS and ITSG to their configuration management plan.

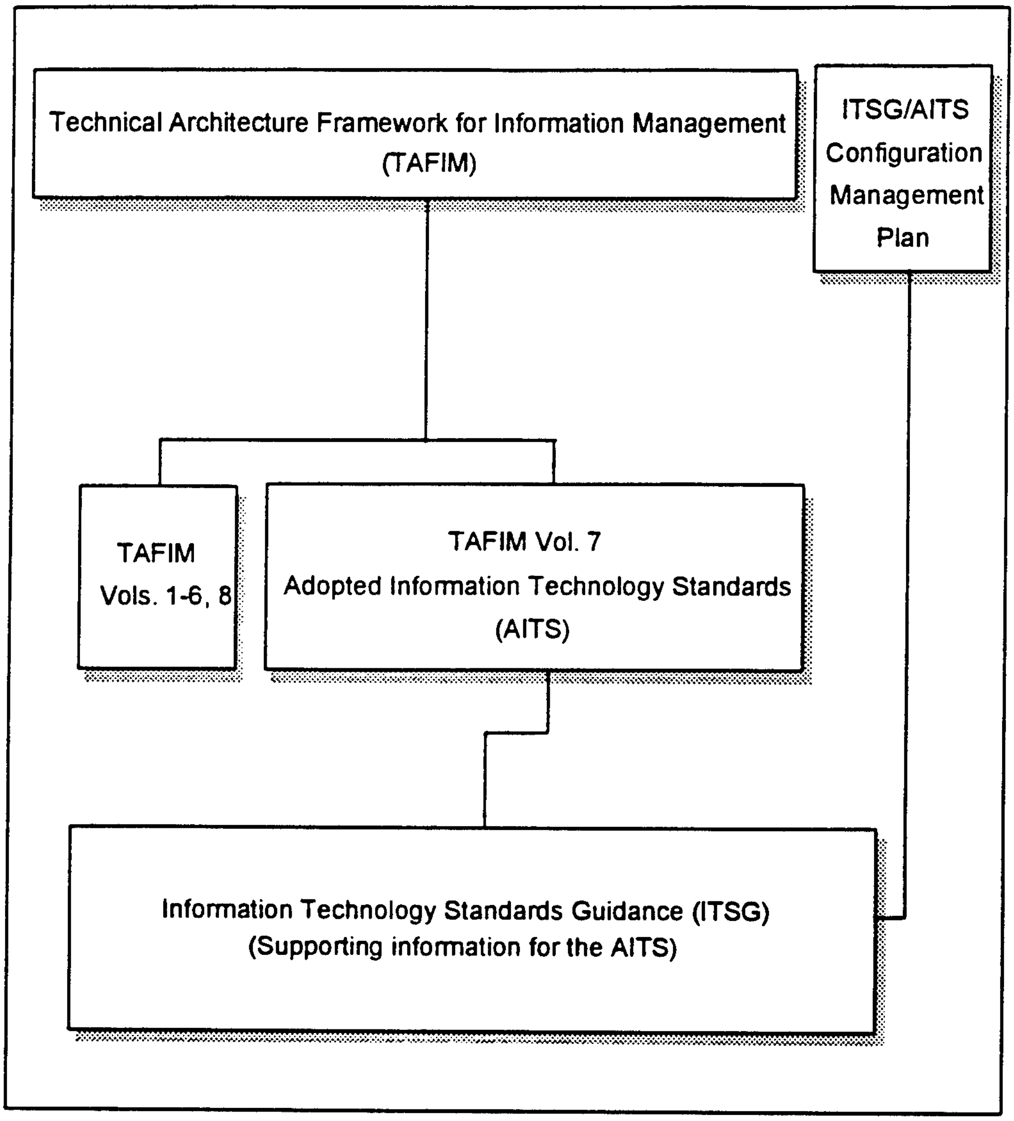

Technical Architecture Framework for Information Management (TAFIM)
ITSG/AITS Configuration Management Plan
TAFIM Vols. 1-6, 8
TAFIM Vol. 7 Adopted Information Technology Standards (AITS)
Information Technology Standards Guidance (ITSG) (Supporting information for the AITS)

Figure 1-1. Relationship of AITS and ITSG to their Configuration Management Plan

1.3 AUDIENCE

The AITS provides adopted DoD standards guidance policy and the ITSG provides amplifying implementation guidance to:

- Organizational policy makers who develop guiding policies
- System managers and resource sponsors who validate requirements
- System architects and planners who identify the functional requirements needed to fulfill the program or system requirements

- Acquisition officials and supporting system engineers who will utilize the AITS in contractual actions

- Implementors who will use the information to assist in development and modernization efforts not supported by system profiles.

1.4 HISTORY

Originally, DoD IT standards guidance was promulgated as a chapter of the DoD Technical Reference Model (TRM). The TRM was based upon the National Institute of Standards and Technology (NIST) product called the Application Portability Profile (APP). DISA/JIEO/CFS had also embarked on an initiative to provide detailed implementation guidance and develop a consensus-based DoD definition of an OSE with the document called *The Open Systems Environment Profile for Imminent Acquisitions (OSE/IA)*. The TAFIM initiative has captured the collective guidance and information of all these efforts and has now integrated and promulgated it as Volume 7 of the TAFIM, the *Adopted Information Technology Standards* (AITS). This consensus standards profile is the product of an extensive coordination and review process regulated by the Defense Standardization Program, per DoD 4120.3-M (Defense Standardization Program Policies and Procedures). Its development was accomplished through the support of multiple technical working groups and comprehensive reviews by the CINCs, Services, and Agencies.

1.5 FUTURE STANDARDS REQUIREMENTS

The technology within the focus of the AITS is growing and changing dynamically. Additionally, the standards organizations are actively adding to the body of consensus-based standards. The emerging internationalization of IT standards requirements is stimulating both harmonization and acceleration of standardization activity to accommodate compatibility and competitiveness in the world arena. To meet the challenges of the fast-paced IT domain and the decentralized decision-making essential to the execution of DoD programs, the AITS and accompanying ITSG are evolving together in a manner consistent with events in standards bodies. They will be published on a regular cycle. CFS, within DISA, is responsible for the evolution of the IT standards policy and is prepared to provide customer assistance in applying the information provided. The consumer of AITS and ITSG information is encouraged to contact CFS for assistance or to identify functional requirements and/or standards not yet incorporated into the document. The CFS will appreciate additional inputs on the use of specific standards, deficiencies, and future needs using the response format found in the appendix.

1.6 AITS DEVELOPMENT AND COORDINATION PROCESS

For an explanation of the coordination and configuration management process for the AITS and ITSG, see the *Adopted Information Technology Standards and Information Technology Standards Guidance Configuration Management Plan* (and also Figure 1-1). Version 2.0 of the AITS was the first to achieve approval by the SCC after several transformations of the format and degree of supporting information presented. It constitutes a baseline for the collective set of

IT standards to be commonly used for DoD systems. Versions 2.1 and 2.5 were created to reflect changes prompted by the most recent review of the AITS and ITSG and changes in standards guidance.

This page intentionally left blank.

2.0 OSE PRINCIPLES AND THE AITS

2.1 OPEN SYSTEM DEFINITION

The AITS and ITSG together comprise a definition of the service areas supported within the IT domain. This domain is then broken down into lower levels, as is explained below. Through the process of standardization upon a consistent and stable framework, it becomes possible to compare and contrast the efficacy of competing standards and to describe functional requirements, assess standardization needs, and support development of profiles.

Each major heading, Major Service Area (MSA), establishes a grouping of services or functionality defined by industry standards and is expressed in a way to be consistent with the manner in which the standards bodies are addressing these groups. The sub-headings, Mid Level Service Area (MLSA) and Base Service Area (BSA), identify more specific, concrete examples of the functionality under the major grouping. The functionality described by the MSAs, MLSAs, and BSAs defines the services available from the application platform across the platform interfaces, application programming interfaces (APIs), and external environment interfaces (EEIs).

The MSA category is the highest level of IT functionality. MSAs provide the overall set of standards services that support the objectives of application portability and system interoperability. The MSAs include Software Engineering Services, User Interface Services, Data Management Services, Data Interchange Services, Graphics Services, and Network Services.

MSAs are divided into areas, called MLSAs, that provide like functionality and further decompose the IT functionality. This decomposition is intended to provide a more precise description of each MSA. The number of categories in each MLSA varies, depending on the variation and complexity of the functionality included in the MSA.

The BSA is the next level of granularity below the MLSA and provides the most precise description of IT functionality in any MSA. The BSAs further decompose the IT functionality in each MLSA category. The BSAs are fully described in the ITSG.

The ITSG extends the current open systems environment (OSE) definition to enable identification of required functions and services, including those that are not yet supported by standards. In Figure 2-1, the decomposition of the ITSG is shown. Primary TRM definition elements are:

- Application Programming Interfaces (APIs)

- External Environment Interfaces (EEIs).

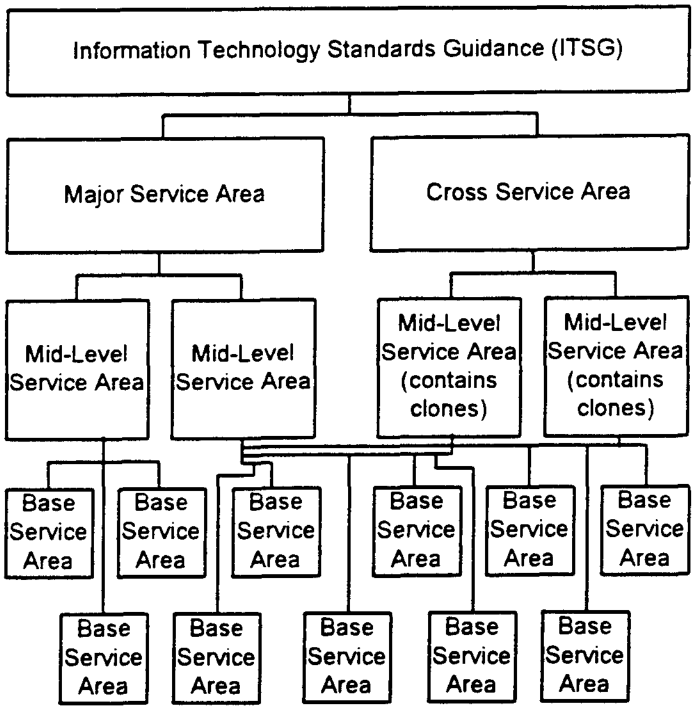


Figure 2-1. Decomposition of the ITSG

The evolving ITSG's OSE definition adds the following elements to complement those OSE elements above:

- Base Service Areas
- Procedural Standards
- Bindings
- Environment Transition Paths.

2.2 STANDARDS AS REQUIREMENTS

Within each program using IT to accomplish system functions, the underlying standards comprise a specialized subset of the OSE Requirements Definition. Standards support the accomplishment of a functional requirement in a manner consistent with common practice, best value, and optimal adaptability to *yet to be identified* requirements. The innovation underway today will be tomorrow's legacy. Effective use of commonly adopted standards to regulate the implementation of definable functions increases the likelihood of adaptability and interoperability throughout the life-cycle of a system or application. However, total expression of OSE requirements using standards is impractical because of the need to specify requirements where no standards exist.

2.3 DoD IT STANDARDS MANAGEMENT

In every instance where there is an identified need for adoption of an open commercial standard to support a DoD requirement, there is an accompanying need to ascertain the appropriate DoD role within the related standardization project.

2.3.1 Standards Leadership and Advocacy Support Role

For those requirements where the technical solution to a DoD requirement falls within the scope of an existing standardization initiative and the technology is relatively mature, it is usually best for DoD's standards representatives to support an existing process and advocate for the unique elements of the DoD requirement. In this way, the DoD requirement becomes aligned with a broadly supported standard and optimizes the opportunity for commercialization of the DoD requirement. The increasing internationalization in the IT market provides greater opportunities for the expression of DoD requirements in a standardization forum where interoperability and compatibility with international allies can be accomplished through open standards.

2.3.2 Product Selection Role

For those requirements where the technical solution is at the forefront of technology, standardization has seldom occurred in time to satisfy the DoD implementation requirement. Clear identification of *best practice* by a standardization organization has not been possible due to the immaturity of the technology and emerging innovations. In these cases, it is sometimes in the best interest to select a most probable *best practice*. This selection must then be supported by an aggressive and effective advocacy throughout the standard's life-cycle by DoD's standards representatives to ensure its adoption in an appropriate open and consensus-regulated standardization body. The life-cycle requirement may motivate escalation of the standardization initiative to an international forum. The preliminary and rapidly evolving definition of the new standard may require specification via mechanisms of lower preference in the hierarchy of standards. As implementations become proven and the technology matures, the DoD goal is to ensure the specification is migrated upward in preference in the hierarchy of standards through the execution of a life-cycle plan for the standard.

This page intentionally left blank.

3.0 ADOPTED INFORMATION TECHNOLOGY STANDARDS

3.1 SUPPORTING PROCESS

The evolution of the service areas and supporting standards in the AITS will be guided by the various working groups which will define DoD requirements and evaluate the technical standardization solutions. The CFS will integrate this effort and ensure its timely promulgation through regular updates to the TAFIM.

3.2 DoD INTEGRATION MODEL

The DoD Integration Model, described in Volume 1 of the TAFIM, is a method for achieving functional and technical integration of business processes and information systems. It describes five integration levels, each building on the preceding level.

- Level 1 is the Enterprise (or DoD-wide) Level. Level 1 encompasses information management (IM) elements that are mandatory across the DoD. It includes IT and IM policy, procedures, standards, and doctrine. This level also includes standard IT capabilities such as technical and data standards; reference models; architectures, methods, tools; and shared computing and communications services. The Enterprise Level standards are represented by the AITS.

- Level 2, the Mission Level, is composed of major DoD mission areas such as Command and Control Systems, Intelligence Systems, and Mission Support or Business Systems. At this level, areas of specialization and functional focus emerge and mandatory DoD-wide technical requirements and capabilities are supplemented with mission area specific requirements and capabilities. Mission Level standards guidance is promulgated in a Mission Area Profile based on the AITS.

- Level 3, the Function Level, breaks the mission areas into the multiple activities and processes of the DoD as identified in DoD 8020.1-M. Architectures and standards are defined for the *to-be* functional practices and processes as based on Mission Level architectures.

- Level 4, the Application Level, includes the development, maintenance and operation of individual information systems. In the integration concept, each mission-area application can support a process, an activity, or a complete function. Individual information system profiles are developed in consonance with the applicable Mission and Function Level profiles.

- Level 5, the Personal Level, includes personal productivity tools and individual tailoring of automated capabilities for the end-users. The tailoring must conform to guidelines and procedures that ensure the integrity of shared resources as well as effective operations.

3.3 STANDARDS SELECTION CRITERIA

The AITS addresses IT standards requirements across DoD. The adoption of one specification from among several addressing a common function requires thorough consideration of several criteria. Crucial tests for inclusion of a specification at the Enterprise Level in support of the OSE goal include the public availability of the specification and the consensus process regulating control of the specification's life-cycle. The following are the key criteria contributing to the selection of a standard for inclusion in the AITS. These criteria are an expansion of the criteria used to evaluate standards within the NIST APP.

- **Meets DoD requirements.** DoD functional requirements will determine the standards that are adopted for DoD use. There is a shift away from military-unique specifications and toward *dual use* of commercial technology. Increasing the use of commercial technologies can lower costs for all concerned. In the case of many *process* and *product* standards, best business practice may also be the optimal solution for DoD, even when 100% of DoD requirements may not be satisfied. However, despite all efforts to identify commercially based specifications, there will continue to be unique military requirements warranting DoD defined specifications.

- **Legal requirements.** Requirements based on the law may specifically mandate the use of specific standards. Automated Data Processing (ADP) standards development was excluded from the Federal Standardization Program in 1965 when Public Law 89-306 (the Brooks Act) established a specific program for standardization of ADP. In addition, the Brooks Act has been amended by Public Law 99-500, which expanded the definition of ADP to include certain aspects of telecommunications previously contained in the Federal Telecommunications Standards Program, and by the Computer Security Act of 1987 (P.L. 100-235). The program to standardize ADP, as defined in these public laws, is carried out by NIST. Mandatory Federal Information Processing Standards (FIPS) are listed in the *Federal ADP and Telecommunications Standards Index*, Doc. No. KMR-94-1-A, published by the General Services Administration.

- **Public specification.** Consistency with the ultimate goal of an OSE is a key criterion in the selection of standards. Some specifications offer a good technical solution, but are not available in an open public forum for potential bidders or developers to utilize.

- **Consensus basis.** The level of consensus, both within industry and across the DoD, is an important consideration. Specifications that are controlled by a single corporate entity, unregulated by a *consensus* processes, are not favored. Acquisition guidance advocates competitiveness in procurement to reduce cost and promote innovation.

- **Product availability.** The degree of market support for specific standards predicts future competitiveness among products implemented upon the standard. Degree of product availability and implementation may influence standard selection on the basis of this criteria.

- **Maturity of technology.** The maturity of the technology and/or the uniqueness of innovative application of a proven technology may impact selection of specific standards.

The standards selected for the DoD profile will represent technologies that have matured to the point where standardization is appropriate but that have not reached a point of obsolescence.

- **Testability.** The ability to validate conformance of an implementation with the specified standard may be crucial to the attainment of the required capabilities. This is especially important for those implementations with interoperability requirements. Standards selected *from* the AITS will be those accompanied by standards which define the procedures by which conformance to the standard are measured. Additional consideration is given to standards which have an existing conformance testing infrastructure in place. There is also a need for test beds to research, describe, and document degrees of interoperability and to perform Operational Test and Evaluation (OT&E) of operational systems to verify the effectiveness and compliance of implemented designs.

- **Internationalization.** Election of one specification over another may be influenced by the extent of internationalization, which includes the ability to accommodate different cultural conventions, character sets, and representations. Requirements for interoperability with allies and foreign suppliers may warrant selection of some specification on the basis of its international sponsor or competitiveness in the international market.

- **Legacy implications.** Compatibility with the installed infrastructure is frequently a requirement. Feasibility of retrofit, adaptation, or other accommodating strategy must be considered. Some specifications may also be selected over others to preserve or sustain process consistency. Many process specifications invoke issues of personnel training and context consistency crucial to sustainment of other processes.

- **Security.** DoDD 5200.28, *Security Requirements for Automated Information Systems (AISs)*, 21 March 1988, specifies minimum security requirements for AIS. Also, procedures for determining minimum AIS computer-based security requirements are described to determine the minimum evaluation class required for an AIS as defined in DoD 5200.28-STD, *DoD Trusted Computer System Evaluation Criteria*, December 1985.

- **Preference.** The preceding criteria constitute technical and economic considerations as described in MIL-STD-970. After consideration of these criteria, standards will be selected for adoption based on a preference list. The selection of a standard or specification of lower preference is to be made only when the standards and specifications of higher preference are not technically or economically suitable for use. The order of preference, from top to bottom, is:

  - Standards mandated by multinational treaty or law

  - Non-government standards

    - Adopted international standards

    - Adopted U.S. non-government standards

  - Other international or U.S. non-government standards
- Commercial item descriptions
- Performance-based Federal specifications or standards
- Performance-based, fully-coordinated military specifications or standards
- Design-based Federal specifications or standards
- Design-based, fully-coordinated military specifications or standards
- Limited coordinated military specifications or standards
- Locally prepared, one-time-use purchase descriptions.

These criteria are used to select a specific standard for DoD adoption. The priority of each standard selection criterion is determined in the context of the specific system standard solution being evaluated. It is important that the selection criteria used in each standard selection be documented and available for use in justifying deviations in evolving the profile as the technology and specifications evolve. In addition, it is important to establish a preferred ordering of specifications within an area to support practical standards-based solutions while accommodating legacy investments. With each system developed, improved, or updated, it is the overarching objective to consistently move closer to a common, practical OSE solution.

3.4 RELATIONSHIP BETWEEN AITS STANDARDS AND WEAPON SYSTEM STANDARDS

The standards in the AITS have a much broader range of applicability than just information processing systems. They are equally applicable to other systems, such as weapon systems. AITS standards in Major Service Areas such as data interchange, operating systems, and security are as needed by many weapon systems as mission critical computer resources (MCCR) standards are. The magnitude of the usability is reflected in Figure 3-1.

Among the Major Service Areas of the AITS that contain standards useful to military weapon systems are user interface (e.g., keyboard device layout, user interface style guides), data management (e.g., data dictionary/directory services), data interchange (e.g., physical interface, image data interchange, geospatial data exchange, tactical communications), graphics (e.g., symbology graphics), networking (e.g., connectionless service), operating systems (e.g., real time services and interfaces), system management (e.g., fault monitoring), and security (e.g., authentication).

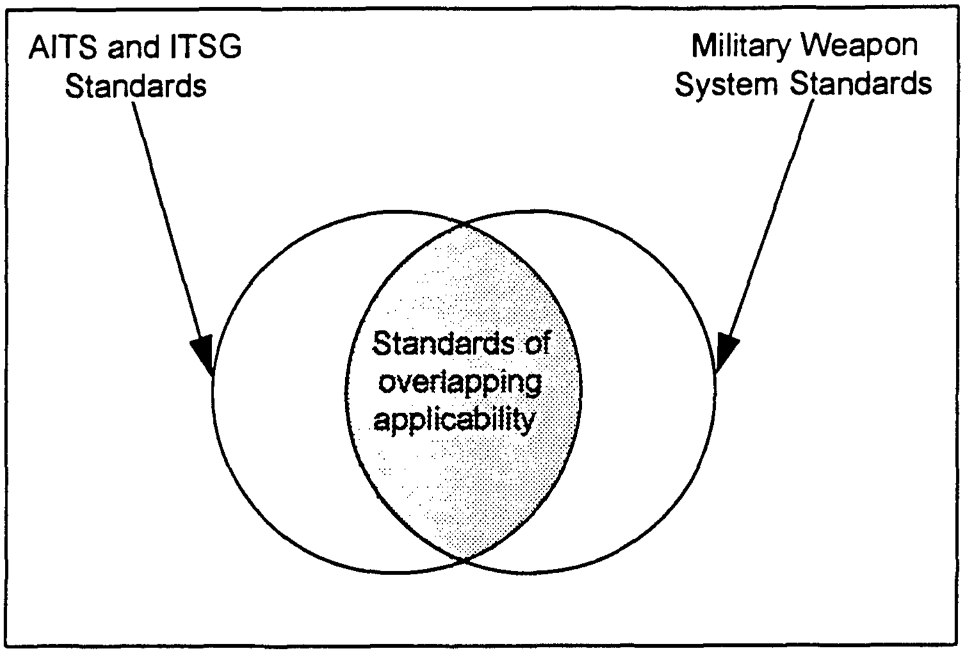

AITS and ITSG Standards
Military Weapon System Standards
Standards of overlapping applicability

Figure 3-1. AITS Standards and Weapon System Standards

3.5 ADOPTED INFORMATION TECHNOLOGY STANDARDS

The AITS are represented in Appendix A as a high-level tabular matrix organized by the MSAs of the DoD TRM. Each Major Service Area in the TRM is represented in the AITS as a collection of MLSAs. MLSA are composed of smaller, defined services called BSAs. A BSA might contain an Adopted Standard selected to meet the functional requirements of the BSA if the adopted standard meets the criteria previously listed.

This page intentionally left blank.

4.0 APPLICATION OF THE ADOPTED INFORMATION TECHNOLOGY STANDARDS

4.1 OVERVIEW

The AITS provides the Enterprise Level OSE guidance which, when applied to DoD systems, will move DoD to an open system environment and facilitate interoperability, transportability, and scalability of applications. Each system must select from and augment the AITS with standards and specifications that apply to the specific functions the system performs. For example, an intelligence system may have a specific set of standards that differs from a finance system, based on required OSE functionalities, but both systems will comply with the AITS.

A system is designed and developed to perform specific functions. The DoD Integration Model, introduced in Section 3.1, is based on the fact that systems within a functional area share many common requirements. Interoperability is enhanced by the use of a common foundation of standards within the functional domain.

The AITS provides the Enterprise Level standards guidance for DoD. All upper level profiles must comply with the AITS selected standards to meet specific system functions. Mission profiles address functional requirements common to a mission domain. Mission domain analysis identifies functionality sets to be supported by standards-based implementations. Mission area profile development leads to the definition of additional functional areas with supporting standards. The process supporting the AITS life-cycle utilizes these standards efforts to generate standardization projects supporting identified needs.

Functional Level profiles provide greater refinement of specific capabilities required to achieve performance objectives. Functional profiles may be applicable in multiple missions and in a repeatable manner throughout the enterprise.

The system designer is encouraged to select standards and specifications for these functions using the amplifying guidance in the ITSG. These standards and specifications together with the standards selected from the AITS will form the system profile. This system profile should be similar to most of the systems within a specific mission and function. Application Level profiles include very detailed standards implementation information.

4.2 INFORMATION TECHNOLOGY STANDARDS GUIDANCE (ITSG)

The ITSG is a companion document to the AITS, containing additional detail necessary for the selection of Mission, Function, and Application-Level standards. The ITSG is divided into the TRM major service areas: software engineering, user interface, graphics, data management, data interchange, network, operating system, system management, security, distributed processing, and internationalization services. The ITSG refines these service definitions, identifying over 350 BSAs in the DoD OSE that might be required in a DoD acquisition. These range from broad areas such as programming languages to detailed services such as shared memory, help screens, and object request broker standards. For each service, the ITSG identifies consensus-based industry and DoD standards, as well as unilaterally controlled specifications. It discusses deficiencies with competing standards and identifies related standards areas. It highlights emerging standards expected to effect pre-planned product improvements or technology insertion. Each service description concludes with a DoD consensus recommendation on the standards to be applied if this OSE service is required.

4.3 RESPONSIBILITIES

4.3.1 DISA Center for Standards (CFS)

DISA CFS has the responsibility for developing standards and standards guidance for the DoD. This guidance is contained in the AITS and the ITSG. Guidance on the use of standards for OSE functions not covered in the AITS is provided in the ITSG.

DISA CFS will provide assistance in the development of profiles. DISA CFS will also certify profiles for compliance with DoD open system guidance.

DISA CFS will maintain a library of all profiles, particularly those at the Mission and Function Level that form the basis for higher-level profiles.

4.3.2 Mission/Functional Area Architects

Based on the requirements of their domain, mission and functional area architects will develop profiles that provide guidance for their levels of the integration model. These profiles will be based on the AITS and, in the case of functional profiles, will be based on the respective mission area profile. The developer will submit the profile to DISA for certification.

4.3.3 System Designers

Systems designers will develop their application profile based on specific system requirements and the relevant functional area profile or profiles.

4.3.4 Acquisition Officials

Acquisition officials will use the profile of standards in contractual actions and ensure that standards required on a contract are consistent with the AITS and the IM integration model hierarchy of profiles.

4.3.5 Implementors

Implementors will baseline their existing systems in preparation for migration to their defined system OSE objectives. Implementors will establish their target OSE definitions and ensure that standards are incorporated in the development and evolution of the system to meet their defined OSE objectives.

APPENDIX A.
THE ADOPTED INFORMATION TECHNOLOGY (AITS)

A.1 AITS FIGURES

MAJOR SERVICE AREA: SOFTWARE ENGINEERING SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
CASE tools and environments	
Software development environment	ANSI/IEEE 1209-1992 (Evaluation and Selection of CASE Tools)
Specialized language and compiler tools	ISO/IEC 9945-2:1993 (POSIX, part 2: Shell and Utilities)
(Alternative)	X/Open C436:1994 (Commands and Utilities)
Software life cycle processes [Pending completion of IEEE 1498/EIA 640, MIL-STD-498 is recommended for use subject to Agency/Service policy. ISO/IEC DIS 12207 Software Life Cycle Processes is currently in the international standardization process.] [In light of DoD's new policy on MIL-STDs, MIL-STD-498 is in the process of becoming an IEEE standard.]	
Software life cycle processes	MIL-STD-498 (Software Development and Documentation)
Configuration management	ANSI/IEEE 828-1990 (Software Configuration Management Plans)
(Complementary)	ANSI/IEEE 1042-1987 (Guide to Software Configuration Management)
	MIL-STD-498 (Software Development and Documentation)
Documentation	MIL-STD-498 (Software Development and Documentation)

MAJOR SERVICE AREA: SOFTWARE ENGINEERING SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Joint reviews	ANSI/IEEE 1028-1988 (Software Reviews and Audits)
(Complementary)	MIL-STD-498 (Software Development and Documentation)
Software requirements	ANSI/IEEE 830-1984 (Guide to Software Requirements Specifications)
(Complementary)	MIL-STD-498 (Software Development and Documentation)
Software design	ANSI/IEEE 1016-1987 (Recommended Practice for Software Design Descriptions)
(Complementary)	ANSI/IEEE 1016.1-1993 (Guide for Software Design Descriptions)
	ANSI/IEEE 990-1987 (Recommended Practices for Ada as a Program Design Language)
	MIL-STD-498 (Software Development and Documentation)
Software management indicators	MIL-STD-498 (Software Development and Documentation)
(Complementary)	ISO/IEC 9126 (Quality Characteristics and Guidelines for their Use)
	ANSI/IEEE 982.1-1988 (Standard Dictionary of Measures to Produce Reliable Software)
	ANSI/IEEE 982.2-1988 (Guide for the Use of Standard Dictionary of Measures to Produce Reliable Software)
	ANSI/IEEE 1045-1992 (Software Productivity Metrics)
	ANSI/IEEE 1061-1992 (Software Quality Metrics Methodology)

MAJOR SERVICE AREA: SOFTWARE ENGINEERING SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Software testing and product evaluation (Complementary)	ANSI/IEEE 829-1983/R1991 (Software Test Documentation)
	ANSI/IEEE 1008-1987 (Software Unit Testing)
	NIST FIPS PUB 132 (Guide for Software Verification and Validation Plans)
	ANSI/IEEE 1012-1987 (Software Verification and Validation Plans)
	ANSI/IEEE 1059-1993 (Guide for Software Verification and Validation Plans)
	MIL-STD-498 (Software Development and Documentation)
Software quality assurance (Complementary - by sponsor)	ISO 9001:1987 (Model for Quality Assurance)
	ISO 9000-3:1991 (Guidelines for Application of ISO 9001)
	ANSI/IEEE 730.1-1989 (Software Quality Assurance Plans)
	IEEE 1298-1992 (Software Quality Management System)
	MIL-STD-498 (Software Development and Documentation)
Software problem categories/priorities (Complementary)	IEEE 1044-1993 (Classification for Software Anomalies)
	MIL-STD-498 (Software Development and Documentation)
Software safety	MIL-STD-882 (System Safety Program Requirements)

MAJOR SERVICE AREA: SOFTWARE ENGINEERING SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Software support (Complementary)	MIL-STD-498 (Software Development and Documentation)
	ANSI/IEEE 1219-1993 (Software Maintenance)
Software distribution	OSF DME: Distributed Services
License management	OSF DME: License Management
Languages	
Ada	ISO/IEC 8652:1995 (Ada95)
(Complementary)	NIST FIPS PUB 119-1 (Ada95)
C	ANSI/ISO 9899: (C)
(Complementary)	NIST FIPS PUB 160
FORTRAN	NIST FIPS PUB 69-1 (FORTRAN-77)
(Alternative)	ISO 1539:1990 (FORTRAN-90)
COBOL	NIST FIPS PUB 21-4 (COBOL)
JOVIAL	MIL-STD-1589C, Notice 1, 1994 (JOVIAL)
MUMPS (aka *M)*	NIST FIPS PUB 125-1 (MUMPS aka M)
Bindings	
Ada bindings	ISO 9075:1992 (Binding to SQL)
(Complementary)	ISO/ANSI 9593-3:1990 (Binding to PHIGS)
	IEEE 1003.5-1992 (POSIX Ada Language Interfaces)
	IEEE 1003.5b (POSIX Ada Real Time Binding)

MAJOR SERVICE AREA: SOFTWARE ENGINEERING SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	ANSI X3.168-1989 (Embedded SQL and SQL Ada Module Extensions)
	NIST FIPS PUB 127-2 (SQL, for Ada bindings)

This page intentionally left blank.

MAJOR SERVICE AREA: USER INTERFACE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
User Interface	
Keyboard device layout	ISO 9995-1..8:1994 (Keyboard Device Layout)
Graphical Client-Server Operations	
Data stream encoding	NIST FIPS PUB 158-1 (X-Windows)
Data stream interface	NIST FIPS PUB 158-1 (X-Windows)
Subroutine foundation library	NIST FIPS PUB 158-1 (X-Windows)
Raster data interchange	MIL-PRF-28002 (CALS Raster)
(Alternative)	NIST FIPS PUB 150 (Group 4 Facsimile)
	NIST FIPS PUB 158-1 (X-Windows, for BDF)
User interface management system	NIST FIPS PUB 158-1 (X-Windows)
Communication between GUI client applications	OSF Motif AES 1.2: ICCCM, v 1.0
Data interchange format for GUI-based applications	OSF Motif AES 1.2: ICCCM, v 1.0
(Complementary)	NIST FIPS PUB 158-1 (X-Windows)
Compound text encoding	X/Open CTE, v1.1
X logical font description	X/Open XLFD, v1.3
Object definition and management	
3-D appearance	NIST FIPS PUB 158-1 (X-Windows, for PEX)
GUI internationalization support	X/Open G304:1993 (Internationalisation Guide)

MAJOR SERVICE AREA: USER INTERFACE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Interchange format for design tools	COSE Motif
Application programming interfaces	IEEE 1295-1993 (Motif)
Language bindings for bit-mapped GUIs	IEEE 1295-1993 (Motif)
Style guide	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
User interface definition language	OSF Motif AES 1.2: UIDL
Window management	
Independent window management services	OSF Motif AES 1.2
Multiple displays	OSF Motif AES 1.2
Style guide	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
Drivability	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
On-line help	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
Commands, menus, and dialog	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
Character-based user interface	
Style guide	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
Electronic forms	JIEO-E-2300 (Electronic Forms Systems)

MAJOR SERVICE AREA: DATA MANAGEMENT SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Database management system	
Basic database services	NIST FIPS PUB 127-2 (SQL)
(Complementary)	NIST FIPS PUB 193 (SQL Environments)
Index sequential access	X/Open D010:1990 (ISAM Developers' Specification)
(Complementary)	X/Open C215:1992 (Data Management, Issue 3: ISAM)
Multidatabase APIs	X/Open P303:1993 (SAG Call Level Interface)
Database administration	DoDD 8320.1 (DoD Data Administration)
Electronic forms	JIEO-E-2300 (Electronic Forms Systems)
Data dictionary/directory services	
Data dictionary	NIST FIPS PUB 156 (IRDS)
Transaction processing	
Protocol for heterogeneous interoperability	ISO 10026-1,2,3:1992 (OSI Distributed Transaction Processing)
Transaction manager-resource manager interface	X/Open C193:1992 (XA Specification)
Transaction demarcation	X/Open P209:1992 (TX Specification)
Transaction manager to communications manager interface	X/Open S423:1994 (XA+ Specification)
(Complementary)	X/Open P306:1993 (XATMI Specification)
	X/Open P305:1993 (TxRPC Specification)

MAJOR SERVICE AREA: DATA MANAGEMENT SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Distributed queuing	IEEE P1003.15 (POSIX Batch Extensions)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Characters and symbols	
Font information exchange	ISO 9541-1,2,3:1991-94 (Font Information Interchange
Hardware applications	
External data representation	ITU-T X.409 (XDR for use with X.400)
Circuit design data exchange	NIST FIPS PUB 172 (VHDL)
Bar coding	MIL-STD-1189B (Standard DoD Bar Code Symbology
Physical interface (Alternative)	NIST FIPS PUB 22-1 (Synchronous Signalling Rates between Data Terminal and Data Communication Equipment)
	NIST FIPS PUB 100-1 (DTE/DCE Interface)
	NIST FIPS PUB 166 (4800/9600 bps 2-wire duplex modems)
	NIST FIPS PUB 167 (9600 bps four-wire duplex modems
	NIST FIPS PUB 168 (12000/14400 bps 4-wire duplex modem)
	NIST FIPS PUB 169 (Error correction in modems)
	NIST FIPS PUB 170 (Data compression in V.42 modems)
	PCMCIA PC Card Standard, Release 2.1
Optical digital technologies	
Read-only optical discs	ISO 9660:1988 (Volume/file structure for CD-ROM)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Write-once optical discs (Complementary by size)	ISO/IEC 9171-1:1990 (Unrecorded 130mm WORM)
	ISO/IEC 9171-2:1990 (Recording format for 130mm WORM)
	ANSI X3.191-1991 (130mm WORM)
	ANSI X3.211-1992 (130mm WORM)
	ANSI X3.214-1992 (130mm WORM)
	ISO/IEC 11560:1992 (130mm WORM using Magneto-Optical Effect)
	ANSI X3.220-1992 (130mm WORM using Magneto-Optical Effect)
	ISO/IEC 10885:1993 (356mm WORM)
	ANSI X3.200-1992 (356mm WORM)
Rewritable optical discs (Complementary by size)	ISO 10900:1992 (90mm Optical Disk, Rewritable and Read Only)
	ISO 10089:1991 (130mm Rewritable Optical Disk)
	ANSI X3.212-1992 (130mm Rewritable Optical Disk Using Magneto-Optical Effect)
Document interchange	
Document exchange (Alternative)	MIL-PRF-28001 (CALS SGML)
	NIST FIPS PUB 152 (SGML)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Custom definition of document types	NIST FIPS PUB 152 (SGML)
Electronic forms interchange	JIEO-E-2300 (Electronic Forms Systems)
Technical data interchange	
Vector graphics data interchange (Alternative)	MIL-PRF-28000 (CALS IGES)
	NIST FIPS PUB 177 (IGES)
	MIL-PRF-28003 (CALS CGM)
	MIL-STD-2301A (NITFS CGM)
	NIST FIPS PUB 128-1 (CGM)
Product data interchange (Alternative on CALS)	MIL-PRF-28000 (CALS IGES)
	NIST FIPS PUB 177 (IGES)
	ISO/IEC 10303:1994 (STEP)
	MIL-STD-1840B (Automated Interchange of Technical Information (CALS))
Business data interchange	NIST FIPS PUB 161-1 (EDI)
Raster/image data interchange	
Raster data interchange (Alternative)	MIL-PRF-28002 (CALS Raster)
	NIST FIPS PUB 150 (Group 4 Facsimile)
	NIST FIPS PUB 158-1 (X-Windows, for BDF)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Image data interchange (Complementary)	MIL-STD-2500A (NITFS, v. 2.0)
	MIL-HDBK-1300A (NITFS)
DoD applications	
Military logistics and document support (Alternative)	MIL-STD-1840B (Automated Interchange of Technical Information (CALS)
	MIL-STD-498 (Software Development and Documentation)
	MIL-STD-1388-2B (LSA Record)
Geospatial data exchange (Alternative)	MIL-STD-2407 (Vector Product Format)
	MIL-STD-2401 (World Geodetic System)
	STANAG 3809 (Digital Terrain Elevation Data)
	STANAG 7074 (Digital Geographic Information Exchange Standard (DIGEST))
	NIST FIPS PUB 173-1 (Spatial Data Transfer Standard)
	MIL-STD-2411 (Raster Product Format)
Symbology graphics (Alternative)	MIL-STD-2525 (Common Warfighting Symbology)
	MIL-STD-2402 (Symbology Standard)
	WMO Document #49 (Meteorological Services)
	MIL-STD-1295A (Helicopter Cockpit Display Symbology)
	MIL-STD-1787B (Aircraft Display Symbology)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Exchange of formatted military messages	Interim MIL-STD-6040 and CJCSM 6120.05 (MTF)
(Alternative)	STANAG 5500 and ADATP 3 (MTF)
	MIL-STD-6011 (TADIL A and B)
	MIL-STD-6004 (TADIL C)
	STANAG 5501 and ADATP 31 (Link 11)
	STANAG 5504 and ADATP 4 (Link 4)
	STANAG 5511 and ADATP 11 (Link 11 and 11B)
	STANAG 5516 and ADATP 16 (Link 16)
	STANAG 5601 and ADATP 12 (Ship-Shore-Ship Buffer)
	MIL-STD-2500A (NITFS, v. 2.0)
	Joint Pub 3-56.20 through 23 (Multi-TADIL Operating Procedure)
	JIEO Multi-TADIL Data Extraction/Reduction Guide
	JTIDS TIDP-TE (TADIL J)
	Interim JTIDS Message Specification (IJMS) Decision Paper 4 and 5
	IJMS Decision Paper 6 (IJMS SOP)
	MIL-STD-6013 (ATDL-1)
	Variable Message Format (VMF) TIDP-TE

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Tactical communications	MIL-STD-2045-44500 (TACO2 for the NITFS)
(Alternative)	MIL-STD-188-203A-1 (TADIL A)
	MIL-STD-188-212 (TADIL B)
	MIL-STD-188-203-3 (TADIL C)
	MIL-STD-188-220 (Digital Message Transfer Device (DMTD)
Continuous Acquisition and Life-Cycle Support (CALS)	MIL-STD-1840B (Automated Interchange of Technical Information (CALS)
(Complementary)	MIL-HDBK-59B (CALS Implementation Guide)
	MIL-M-87268 (IETM General)
	MIL-D-87269 (Database Revisable IETM)
	MIL-Q-87270 (IETM Quality Assurance)
	MIL-STD-974 (Contractor Integrated Technical Information Service - CITIS)
Compression	
Text and data compression	X/Open C436:1994 (Commands and Utilities)
Still image compression	NIST FIPS PUB 147 (Group 3 Compression)
(Alternative)	NIST FIPS PUB 148 (General Facsimile)
	NIST FIPS PUB 150 (Group 4 Facsimile)
	ITU-T T.4-1988 (Group 3 Compression)
	ITU-T T.6-1988 (Group 4 Compression)

MAJOR SERVICE AREA: DATA INTERCHANGE SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	ITU-T T.81-1993 (JPEG)
	MIL-STD-188-196 (NITFS Bi-Level)
	MIL-STD-188-197A (NITFS ARIDPCM)
	MIL-STD-188-198A (NITFS JPEG)
	MIL-STD-188-199 (NITFS Vector Quantization)
	ISO/IEC 10918-1 (JPEG)
Motion image compression	ISO 11172-1,2,3:1993 (MPEG)

This page intentionally left blank.

MAJOR SERVICE AREA: GRAPHICS SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Raster graphics	
Raster data interchange	MIL-PRF-28002 (CALS Raster)
(Alternative)	NIST FIPS PUB 150 (Group 4 Facsimile)
	NIST FIPS PUB 158-1 (X-Windows, for BDF)
Still image compression	NIST FIPS PUB 147 (Group 3 Compression)
(Alternative)	NIST FIPS PUB 148 (General Facsimile)
	NIST FIPS PUB 150 (Group 4 Facsimile)
	ITU-T T.4-1988 (Group 3 Compression)
	ITU-T T.6-1988 (Group 4 Compression)
	ITU-T T.81-1993 (JPEG)
	MIL-STD-188-196 (NITFS Bi-Level)
	MIL-STD-188-197A (NITFS ARIDPCM)
	MIL-STD-188-198A (NITFS JPEG)
	MIL-STD-188-199 (NITFS Vector Quantization)
	ISO/IEC 10918-1 (JPEG)
Vector graphics	
Vector graphics API	NIST FIPS PUB 153 (PHIGS)
(Complementary)	ISO/IEC 9592-4:1992 (PHIGS PLUS)
Vector graphics data interchange	MIL-PRF-28000 (CALS IGES)
(Alternative)	NIST FIPS PUB 177 (IGES)
	MIL-PRF-28003 (CALS CGM)

MAJOR SERVICE AREA: GRAPHICS SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	MIL-STD-2301A (NITFS CGM)
	NIST FIPS PUB 128-1 (CGM)
Device interfaces	
Device interface API	ISO/IEC 9636-1..6:1991 (CGI)

MAJOR SERVICE AREA: COMMUNICATIONS SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Application services	
File transfer	MIL-STD-2045-17504 (FTP)
Remote file access	OSF DCE 1.1: DFS
Message transfer	ANSI/IEEE 1224.1 (X.400 E-mail API)
(Complementary)	ACP 123
	ACP 123 US SUPP-1
Terminal emulation	MIL-STD-2045-17506 (Remote Login Profile)
Remote login	MIL-STD-2045-17506 (Remote Login Profile)
Remote procedure call	OSF DCE 1.1: RPC
Directory services	ITU-T X.500/01/09/11/18/19/20/21/25
(Complementary)	ANSI/IEEE 1224.2 (Directory/Name Space API)
	ISO 8822, 8823, 8326, 8327
	MIL-STD-2045-17505 (DNS) (legacy systems)
Addressing	ITU-T X.500:1993 (OSI Directory (ISO 9594))
(Alternative)	ISO 8823, 8327
	IEEE 802.2 (1992)
	MIL-STD-2045-14502-1A/4/5 (Internet Transport Profile)
Protocol for interoperability in heterogeneous transaction processing systems	ISO 10026-1, 2,3:1992 (OSI Distributed Transaction Processing)

MAJOR SERVICE AREA: COMMUNICATIONS SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Connection establishment/release	ISO 8823, 8327
(Alternative)	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
	X/Open C303 (XAP)
	IEEE P1003.1g (POSIX protocol - Independent Tranport Service)
	MIL-STD-2045-14503 (RFC 1006)
Connectionless service	ISO 9576/9548 (Connectionless Presentation/Session Protocol)
(Alternative)	MIL-STD-2045-14502-1A/4 (Internet Transport Profile)
	IEEE P1003.1g (POSIX protocol - Independent Tranport Service)
	IEEE 802.2 Type I (1992)
Translation	RFC 1327/1495 (SMTP to X.400 gateway)
(Alternative)	MIL-STD-187-700A
Transport services	
Routing/Relay	MIL-STD-2045-13501
Network gateways	MIL-STD-188-105 (per MIL-STD-187-700A)
Network error recovery	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
Network flow control	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
Network sequencing	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
Priority/precedence	MIL-STD-2045-14502-1A (Internet Transport Profile)

MAJOR SERVICE AREA: COMMUNICATIONS SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Distributed timing service	OSF DCE 1.1
Multicast	ITU-T X.6 (Multicast)
(Alternative)	MIL-STD-2045-14502-1A (Internet Transport Profile)
Subnetwork technologies	
CSMA/CD	MIL-STD-187-700A
(Alternative)	MIL-STD-2045-14502-4/5 (Internet Transport Profile)
Token bus	MIL-STD-187-700A
Token ring	MIL-STD-187-700A
Distributed queue dual bus (DQDB)	MIL-STD-187-700A
FDDI (Fiber optic)	MIL-STD-187-700A
Integrated services digital networks (ISDN)	MIL-STD-187-700A
LAPB	MIL-STD-2045-14502-2 (Internet Transport Profile)
DDN X.25	MIL-STD-2045-14502-3 (Internet Transport Profile)
Frame relay	MIL-STD-187-700A

MAJOR SERVICE AREA: COMMUNICATIONS SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Asynchronous transfer mode (ATM)	MIL-STD-187-700A
Combat net radio digital subnetwork	MIL-STD-188-220A (Digital Message Transfer Device (DMTD)
(Complementary)	MIL-STD-2045-14502-6A (Internet Transport Profile)
Secondary imagery transmission	MIL-STD-2045-44500

MAJOR SERVICE AREA: OPERATING SYSTEM SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Kernel operations	
File management services	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Input/output control	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
System operator services	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Process management and core operating system services	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Environment services	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Hardware error and event conditions	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
System resource limits	NIST SP 500-224 (OIW SIAs for OSEs)
Message queues	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Login services	X/Open C434, C435, C436 (Single UNIX Specification)
Storage device management	OSF DCE 1.1: DFS
Threads interface	OSF DCE 1.1: Threads

MAJOR SERVICE AREA: OPERATING SYSTEM SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
(Alternative)	IEEE 1003.1c (POSIX Threads Extension)
Threads extension language binding	NIST SP 500-224 (OIW SIAs for OSEs)
Kernal language bindings	IEEE 1003.1b:1993, 1003.1g
(Alternatives complementary to FIPS 151-2)	NIST FIPS PUB 151-2 (POSIX.1)
	IEEE 1003.5-1992 (POSIX Ada Language Interfaces)
	IEEE 1003.9 (POSIX FORTRAN Binding)
Media handling	
Backup and restore	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	NIST FIPS PUB 189 (POSIX.2)
Floppy disk format and handling	NIST FIPS PUB 189 (POSIX.2)
Data interchange format	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Shell and utilities	
Commands and utilities	NIST FIPS PUB 189 (POSIX.2)
Print management	NIST FIPS PUB 189 (POSIX.2)
(Alternative)	ISO 10175 (Document Printing Application)
Language bindings to POSIX.2	NIST FIPS PUB 189 (POSIX.2)
Shell programming language	NIST FIPS PUB 189 (POSIX.2)
User-oriented commands and utilities	NIST FIPS PUB 189 (POSIX.2)

MAJOR SERVICE AREA: OPERATING SYSTEM SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
File and program editing services	NIST FIPS PUB 189 (POSIX.2)
Batch scheduling	NIST FIPS PUB 189 (POSIX.2)
Real time extensions	
Memory management	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Scheduling	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
(Complementary)	NIST FIPS PUB 151-2 (POSIX.1)
Semaphores	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Asynchronous I/O	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Asynchronous event notification	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Synchronized I/O	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Real time file system	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
POSIX.1b language bindings	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Fault management services	
Fault management	NMF Omnipoint 1
Clock/calendar services	
Clocks and timers	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Real time timers	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Distributed timing service	OSF DCE 1.1: DTS

MAJOR SERVICE AREA: OPERATING SYSTEM SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Operating system object services	
Object request broker	CORBA Specification Rev. 2.0, 1994

MAJOR SERVICE AREA: SYSTEM MANAGEMENT SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
State management	
Independent window management services	OSF Motif AES 1.2
Batch scheduling	NIST FIPS PUB 189 (POSIX.2)
Process management and core operating system services	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
System administration and management APIs	NIST SP 500-224 (OIW SIAs for OSEs)
(Alternative)	NMF Omnipoint 1
	IEEE 1224
	X/Open C206 (XMP)
Scheduling	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
(Complementary)	NIST FIPS PUB 151-2 (POSIX.1)
User/Group management	
User/Group identification	IEEE P1387.3
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Configuration control	
Software configuration management	ANSI/IEEE 828-1990 (Software Configuration Management Plans)
(Complementary)	ANSI/IEEE 1042-1987 (Guide to Software Configuration Management)

MAJOR SERVICE AREA: SYSTEM MANAGEMENT SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
	MIL-STD-498 (Software Development and Documentation)
Data dictionary	NIST FIPS PUB 156 (IRDS)
System configuration	NMF Omnipoint 1
Network configuration management	NMF Omnipoint 1
Usage management and cost allocation	
Accounting management	NIST FIPS PUB 96
Performance management	
Software management indicators	MIL-STD-498 (Software Development and Documentation)
(Complementary)	ISO/IEC 9126 (Quality Characteristics and Guidelines for their Use)
	ANSI/IEEE 982.1-1988 (Standard Dictionary of Measures to Produce Reliable Software)
	ANSI/IEEE 982.2-1988 (Guide for the Use of Standard Dictionary of Measures to Produce Reliable Software)
	ANSI/IEEE 1045-1992 (Software Productivity Metrics)
	ANSI/IEEE 1061-1992 (Software Quality Metrics Methodology)
Performance management	NIST FIPS PUB 144
(Complementary)	NMF Omnipoint 1
Network flow control	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)

MAJOR SERVICE AREA: SYSTEM MANAGEMENT SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Network sequencing	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
Communication of management information	MIL-STD-2045-38000
Managed information base	MIL-STD-2045-38000
Input/output control	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Event management	NMF Omnipoint 1
(Alternative)	NIST SP 500-224 (OIW SIAs for OSEs)
Fault management	
Software safety	MIL-STD-882 (System Safety Program Requirements)
Network error recovery	MIL-STD-2045-14502-1A/2/3 (Internet Transport Profile)
Fault management	NMF Omnipoint 1
Storage device management	OSF DCE 1.1: DFS
Backup and restore	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	NIST FIPS PUB 189 (POSIX.2)
Hardware error and event conditions	NIST FIPS PUB 151-2 (POSIX.1)
(Complementary)	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
Error and event logging	NMF Omnipoint 1

MAJOR SERVICE AREA: SYSTEM MANAGEMENT SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Other management services	
Database administration	DoDD 8320.1 (DoD Data Administration)
Floppy disk format and handling	NIST FIPS PUB 189 (POSIX.2)
Print management	NIST FIPS PUB 189 (POSIX.2)
(Complementary)	ISO 10175 (Document Printing Application)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Architectures and applications	
System development security	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD NCSC-TG-005, v1 (TNI)
	DoD NCSC-TG-006, v1 (CM in Trusted Systems)
	DoD NCSC-TG-021, v1 (TDI)
	OSF DCE 1.1: Security
	NIST FIPS PUB 151-2 (POSIX.1)
	MIL-STD-498 (Software Development and Documentation)
Database security	NIST FIPS PUB 127-2:1993 (SQL)
	NIST FIPS PUB 156 (IRDS)
Network security architecture	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD NCSC-TG-005, v1 (TNI)
	ISO 10181-2:1993 (OSI Authentication Framework)
	NIST SP 500-224, pt 12,13 (OIW SIAs for OSEs)
	ISO 10745:1993 (OSI Upper Layer Security Model)
	ISO 11586-1:1994 (GULS, part 1)
Operating system security	DoD 5200.28-STD (TCSEC)
(Complementary)	DDS-2600-5502-87 (CMW Security Requirements)
	DDS-2600-6243-92 (CMW Evaluation Criteria)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	NIST FIPS PUB 151-2 (POSIX.1)
	NIST FIPS PUB 112 (Password Usage)
System management security	
Privacy act	PL 100-235 (Computer Security Act of 1987)
(Complementary)	PL 93-579 (Privacy Act of 1974)
Certification and accreditation	DoD 5200.28-STD (TCSEC)
Security risk management	DoD 5200.28-STD (TCSEC)
(Complementary)	NIST FIPS PUB 191 (Guideline for LAN Security)
Security management	ISO 9595, AM4 (CMIS Access Control)
(Complementary)	ISO 10164-7 (System Management Security Alarm Reporting)
	ISO 10164-8 (System Management Security Audit Trail Function)
	ITU-T X.518 (OSI Directory-Distributed Operations)
	DoD 5200.28-STD (TCSEC)
	ISO 9596-1 (CMIP)
	DoD NCSC-TG-005, v1 (TNI)
	DoD NCSC-TG-021, v1 (TDI)
	NMF Omnipoint 1

MAJOR SERVICE AREA: SECURITY SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
	IEEE 1003.1b:1993 (POSIX Real-Time Extensions)
	NIST FIPS PUB 151-2 (POSIX.1)
Security association and key management	NIUF ISDN Security Protocol 421 (SAMP)
(Complementary)	ISO 11586-1:1994 (GULS, part 1)
	ISO 11586-2 (GULS, part 2)
	ISO 11586-3 (GULS, part 3)
	NIST FIPS PUB 171 (Key Management Using ANSI X9.17)
Security audit	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD NCSC-TG-005, v1 (TNI)
	NMF Omnipoint 1
	ISO 10164-8 (System Management Security Audit Trail Function)
Security alarm reporting	ISO 10164-7 (System Management Security Alarm Reporting)
(Complementary)	NMF Omnipoint 1
Authentication	
Personal authentication	DoD 5200.28-STD (TCSEC)
(Complementary)	NIST FIPS PUB 112 (Password Usage)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	NIST FIPS PUB 48 (Automated Personal ID)
	ISO 9594-8.2 (OSI Directory Authentication Framework)
Network authentication	MIL-STD-2045-18500 (MHS Message Security Protocol (MSP) Profile)
(Complementary)	ITU-T X.509 (OSI Directory Authentication Framework)
	DoD NCSC-TG-005, v1 (TNI)
	NIST FIPS PUB 186 (DSS)
	NIST FIPS PUB 180-1 (SHS)
	ISO 8649 (OSI Service Definition for ACSE)
	ISO 8650 (OSI Protocol Specification for ACSE)
	ISO 11586-1:1994 (GULS, part 1)
	ISO 11586-2 (GULS, part 2)
	ISO 11586-3 (GULS, part 3)
	ISO 11586-4 (GULS, part 4)
	IEEE 802.10B-1992 (SILS Secure Data Exchange)
Entity authentication	NIST FIPS PUB 113 (Computer Data Authentication)
(Complementary)	DoD 5200.28-STD (TCSEC)
	ISO 9807 (Retail Message Authentication)
	ISO 9798-1 (Entity Authentication Mechanism)
	ISO 9798-3 (Entity Authentication Mechanism)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Access control	
System access control	DoD 5200.28-STD (TCSEC)
(Complementary)	ISO 9595, AM4 (CMIS Access Control)
Network access control	ISO 9595, AM4 (CMIS Access Control)
(Complementary)	MIL-STD-2045-18500 (MHS Message Security Protocol (MSP) Profile)
	DoD NCSC-TG-005, v1 (TNI)
	IEEE 802.10B-1992 (SILS Secure Data Exchange)
Confidentiality	
Open systems confidentiality	DoD 5200.28-STD (TCSEC)
(Complementary)	PL 93-579 (Privacy Act of 1974)
	PL 100-235 (Computer Security Act of 1987)
Data encryption security	NIST FIPS PUB 46-2 (DES)
(Complementary)	NIST FIPS PUB 74 (Guidelines for DES)
	NIST FIPS PUB 81 (DES Modes of Operation)
	NIST FIPS PUB 185 (EES)
	NIST FIPS PUB 140-1 (Security Requirements for Cryptographic Modules)
	ISO 8372 (Modes of Operation for a 64-Bit Block Cipher Algorithm)
Traffic flow confidentiality	ISO 11577:1994 (NLSP)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Integrity	
Open systems integrity (Complementary)	DoD 5200.28-STD (TCSEC)
	DoD NCSC-TG-021, v1 (TDI)
Data integrity techniques (Complementary)	NIST FIPS PUB 46-2 (DES)
	NIST FIPS PUB 74 (Guidelines for DES)
	NIST FIPS PUB 81 (DES Modes of Operation)
	NIST FIPS PUB 185 (EES)
	NIST FIPS PUB 140-1 (Security Requirements for Cryptographic Modules)
	ISO 8372 (Modes of Operation for a 64-Bit Block Cipher Algorithm)
	NIST FIPS PUB 180-1 (SHS)
	NIST FIPS PUB 186 (DSS)
Network integrity (Complementary)	ISO 11586-1:1994 (GULS, part 1)
	ISO 11586-4 (GULS, part 4)
	IEEE 802.10B-1992 (SILS Secure Data Exchange)
	ITU-T X.500:1993 (OSI Directory (ISO 9594)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Non-repudiation	
Open systems non-repudiation	MIL-STD-2045-18500 (MHS Message Security Protocol (MSP) Profile)
(Complementary)	NIST FIPS PUB 186 (DSS)
	ISO 11586-1:1994 (GULS, part 1)
	ISO 11586-4 (GULS, part 4)
Electronic signature	NIST FIPS PUB 186 (DSS)
Electronic hashing	NIST FIPS PUB 180-1 (SHS)
Availability	
Detection and notification	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD NCSC-TG-005, v1 (TNI)
Security recovery	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD NCSC-TG-005, v1 (TNI)
Security labeling	
User interface security labeling	DoD 5200.28-STD (TCSEC)
(Complementary)	DoD HCI Style Guide, v. 3.0; TAFIM Vol. 8
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	DoDIIS Style Guide
Data management security labeling (Complementary)	DoD 5200.28-STD (TCSEC)
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
Data interchange security labeling (Complementary)	DoD 5200.28-STD (TCSEC)
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
	MIL-STD-2045-48501 (Common Security Label (CSL)
	ITU-T X.411 (MHS Message Transfer System: Abstract Service Definition and Procedures)
Graphics security labeling (Complementary)	DoD 5200.28-STD (TCSEC)
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
Data communications security labeling (Complementary)	MIL-STD-2045-48501 (Common Security Label (CSL))
	DoD 5200.28-STD (TCSEC)
	DDS-2600-6243-92 (CMW Evaluation Criteria)

MAJOR SERVICE AREA: SECURITY SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
Operating system security labeling (Complementary)	DoD 5200.28-STD (TCSEC)
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)
Distributed computing security labeling (Complementary)	DoD 5200.28-STD (TCSEC)
	DoD NCSC-TG-005, v1 (TNI)
	DoD NCSC-TG-021, v1 (TDI)
	DDS-2600-6243-92 (CMW Evaluation Criteria)
	DDS-2600-6243-91 (CMW Labeling Guidelines)
	DDS-2600-6216-91 (CMW Labeling Encoding Format)

This page intentionally left blank.

MAJOR SERVICE AREA: DISTRIBUTED COMPUTING SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Client/server	
Threads	IEEE 1003.1c (Threads Extension to POSIX)
(Alternative)	OSF DCE 1.1: Threads
Remote procedure call	OSF DCE 1.1: RPC
Distributed file service	OSF DCE 1.1: DFS
Naming services	OSF DCE 1.1: Cell Directory Service / Global Directory Service
Distributed timing service	OSF DCE 1.1: DTS
Object services	
Object request broker	OMG CORBA 2.0
Remote access	
File transfer	MIL-STD-2045-17504 (FTP)
Remote login	MIL-STD-2045-17506 (Remote Login Profile)
Remote data access	ISO/IEC 9579-1,2:1993 (RDA)

This page intentionally left blank.

MAJOR SERVICE AREA: INTERNATIONALIZATION SERVICES	
Mid and Base Service Areas (Indented)	**Adopted Standard or Specification**
Character set and data representation	
Coded character sets	ISO 6937:1994 (Coded Character Sets for Text Communication)
7-Bit coded character sets	NIST FIPS PUB 1-2 (Code for Information Interchange)
(Complementary)	ISO 646:1991 (ISO 7-Bit Coded Character Set for Information Exchange)
8-Bit coded character sets	ISO 4873:1991 (ISO 8-Bit Code for Information Interchange)
8-Bit single byte character sets	ISO 8859:1989 (ISO 8-Bit Single-Byte Coded Graphic Character Sets)
Control functions	ISO 6429:1992 (Control Functions for ISO 7-Bit and 8-bit Coded Character Sets)
Code extension techniques	ISO 2022:1986 (ISO 7-Bit and 8-Bit Coded Character Sets - Code Extension Techniques)
Universal character sets	ISO 10646-1:1993 (Universal Multiple-Octet Coded Character Set)
Currency and funds representation	ISO 4217:1990 (Codes for the Representation of Currencies and Funds)
Date and time representation	NIST FIPS PUB 4-1 (Representation of Calendar Date and Ordinal Date)
(Complementary)	NIST FIPS PUB 58-1 (Representation of Local Time of Day)
	NIST FIPS PUB 59 (Representations of Universal Time, Local Time Differentials, and US Time Zone References)
Country name representation	TBD

MAJOR SERVICE AREA: INTERNATIONALIZATION SERVICES

Mid and Base Service Areas (Indented)	Adopted Standard or Specification
Representation of human sexes	TBD
Representation of names of languages	TBD
Cultural convention services	
Numerical value representation	TBD
Customization to local norms	X/Open G304 (Internationalisation Guide, Version 2)
(Complementary)	DOD HCI Style Guide
Natural language support services	
Keyboard device layout	ISO 9995-1..8:1994 (Keyboard Device Layout
Related standards and programs	
Character set registration	TBD

A.2 INDEX OF SERVICE AREAS

The following list is an index of the service areas of the AITS, in alphabetical order. MLSAs appear in *italics*. MSAs appear in **bold**.

A.3 INDEX OF STANDARDS

The following is a list of the standards in the AITS table, consolidated and listed in alphanumeric order.

APPENDIX B

GLOSSARY

Base Service Area (BSA): The lower level of granularity below the Mid Level Service Area which provides the most precise description of IT functionality in any Major Service Area. The BSAs further decompose the IT functionality in each Mid Level Service Area category.

Consensus based: Making decisions based on the agreement of a large majority of the participants.

Major Service Area (MSA): The highest level of IT functionality. MSAs provide the overall set of standards services that support the objectives of application portability and system interoperability.

Mid Level Service Area (MLSA): A division of the MSA that provides like functionality and further decomposes the IT functionality. This decomposition is intended to provide a more precise description of each MSA. The number of categories in each Mid Level Service Area varies, depending on the variation and complexity of the functionality included in the MSA.

Open Systems Environment (OSE): A comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of application, data or people, as specified by information technology standards and profiles.

Profile: A set of one or more base standards, along with specific subsets, classes, options, and parameters, necessary for accomplishing a particular function.

Publicly Available: Available to public without restriction to anyone for implementation, sublicensing, and distribution (i.e., sale) of that implementation.

Specifications: A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behavior, or characteristics of a system or system component. The term is also used to identify additional information that augments a standard.

Sponsor: An advocate for a specific standard or section of a standard who provides significant resources toward the development of the standard.

Standard Selection Criteria: Criteria used in the selection of standards for a profile.

Standard: A document, established by consensus and approved by a government or non-government standards body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the achievement of the optimum degree of order and consistency in a given context.

This page intentionally left blank.

APPENDIX C

ACRONYMS

ACSE	Association Control Service Element
ADATP	Allied Data Transfer Protocol
ADP	Automated Data Processing
AES	Application Environment Specification
AIS	Automated Information System
AITS	Adopted Information Technology Standards
ANSI	American National Standards Institute
API	Application Program Interface
APP	Application Portability Profile
ARIDPCM	Adaptive Recursive Interpolative Pulse Code Modulation
ATDL	Army Tactical Data Link
ATM	Asynchronous Transfer Mode

BDF	Bitmap Distribution Format
BPS	Bits per Second
BSA	Base Service Area

C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CALS	Continuous Acquisition and Lifecycle Support
CASE	Computer Aided Software Engineering
CFS	Center for Standards
CGI	Computer Graphics Interface
CGM	Computer Graphics Metafile
CINC	Commander in Chief
CITIS	Contractor Integrated Technical Information Service
CJCS	Chairman of the Joint Chiefs of Staff
CJCSI	CJCS Instruction
CJCSM	CJCS Manual
CM	Configuration Management
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CMP	Configuration Management Plan
CMW	Compartmented Mode Workstation

CORBA	Common Object Request Broker Architecture
COSE	Common Open System Environment
CSL	Common Security Label
CTE	Compound Text Encoding
DCE	Data Circuit-Terminating Equipment
DCE	Distributed Computing Environment
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DFS	Distributed File System
DIGEST	Digital Geographic Information Exchange Standard
DIS	Draft International Standard
DISA	Defense Information Systems Agency
DME	Distributed Management Environment
DMTD	Digital Message Transfer Device
DNS	Distributed Name Service
DoD	Department of Defense
DoDD	DoD Directive
DoDI	DoD Instruction
DoDIIS	DoD Intelligence Information Systems
DQDB	Distributed Queue Dual Bus
DSS	Digital Signature Standard
DTE	Data Terminal Equipment
EDI	Electronic Data Interchange
EEI	External Environment Interface
EES	Escrowed Encryption Standard
EIA	Electonics Industries Association
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GUI	Graphical User Interface
GULS	Generic Upper Layer Security
HCI	Human-Computer Interface
HDBK	Handbook
ICCCM	Inter Client Communication Conventions Manual

IEC	International Electrotechnical Commission
IEEE	Institute for Electrical and Electronics Engineers
IETM	Interactive Electronic Technical Manual
IGES	Initial Graphics Exchange System
IJMS	Interim JTIDS Message Specification
IM	Information Management
IRDS	Information Resources Directory System
ISAM	Indexed Sequential Access Method
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISP	International Standardized Profile
ISP	ISDN Security Protocol
IT	Information Technology
ITSG	Information Technology Standards Guidance
ITU-T	International Telecommunications Union- Telecommunications
JIEO	Joint Interoperability and Engineering Organization
JPEG	Joint Photographic Experts Group
JTIDS	Joint Tactical Information Distribution System
LAN	Local Area Network
LCM	Life Cycle Management
LIS	Language Independent Specification
LSA	Logistic Support Analysis
MCCR	Mission Critical Computer Resources
MHS	Message Handling System
MLSA	Mid Level Service Area
MNS	Mission Needs Statement
MPEG	Motion Picture Experts Group
MSA	Major Service Area
MSP	Message Security Protocol
MTF	Message Transfer Format
NCSC	National Computer Security Center
NIST	National Institute of Standards and Technology
NITFS	National Imagery Transmission Format Standard
NIUF	National ISDN Users' Forum
NLSP	Network Layer Security Protocol

NMF	Network Management Forum
OIW	OSE Implementors Workshop
ORD	Operational Requirements Document
OSE	Open Systems Environment
OSE/IA	OSE Profile for Imminent Acquisitions
OSF	Open Software Foundation
OSI	Open Systems Interconnection
OT&E	Operational Test and Evaluation
PCMCIA	Personal Computer Memory Card International Association
PEX	PHIGS Extensions to X
PHIGS	Programmer's Hierarchical Interactive Graphics System
PMP	Program Management Plan
POC	Point of Contact
POSIX	Portable Operating Systems Interface for Computers
PUB	Publication
RDA	Remote Data Access
RFC	Request for Comment
RPC	Remote Procedure Call
SAG	SQL Access Group
SAMP	Security Association Management Protocol
SCC	Standards Coordinating Committee
SGML	Standard Generalized Markup Language
SHS	Secure Hashing Standard
SIA	Stable Implementation Agreement
SILS	Standards for Interoperable LAN Security
SMTP	Simple Mail Transfer Protocol
SOP	Standing Operating Procedures
SP	Special Publication
SQL	Structured Query Language
SSL	Standard Security Label
STANAG	Standardization Agreement
STD	Standard
STEP	Standard for the Exchange of Product Model Data

TACO	Tactical Communication Protocol
TADIL	Tactical Digital Information Link
TAFIM	Technical Architecture Framework for Information Management
TBD	To Be Determined
TCSEC	Trusted Computer Systems Evaluation Criteria
TDI	Trusted Database Interpretation
TEMP	Test and Evaluation Master Plan
TIDP-TE	Technical Interface Design Plan - Test Edition
TLSP	Transport Layer Security Protocol
TNI	Trusted Network Interpretation
TRM	Technical Reference Model
TX	Transaction Demarcation
TxRPC	Transactional Remote Procedure Call
UIDL	User Interface Definition Language
VHDL	VHSIC Hardware Description Language
VMF	Variable Message Format
WMO	World Meteorological Organization
WORM	Write-Once Read Many
XA	X/Open Architecture
XATMI	X/Open Application to Transaction Manager Interface
XA+	X/Open Architecture Plus
XDR	External Data Representation
XLFD	X Logical Font Description

This page intentionally left blank.

APPENDIX D

PROPOSING CHANGES TO THE AITS

D.1 INTRODUCTION

This appendix provides guidance for submission of proposed AITS changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of the AITS.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section D.2. Guidance on the use of the format is provided in Section D.3.

The preferred method of proposal receipt is via e-mail in ASCII format, sent via the internet. If not e-mailed, the proposed change, also in the format shown in Section D.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for sending change proposals is shown below.

Internet: stantonj@ncr.disa.mil, with a copy to tafim@bah.com

Mail: Information Processing Directorate

DISA/JIEO/CFS/JEBE (John Stanton)

10701 Parkridge Blvd

Reston, Virginia 22091-4398

Fax: (703) 735-3257; indicate "AITS" on cover sheet

D.2 AITS CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number:

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change #1

(1) Section Number

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording for Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change #2

(1) Section Number

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording for Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

e. **Proposed Change #n**

(1) Section Number

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording for Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

D.3 FORMAT GUIDANCE

The format in Section D.2 should be followed exactly as shown. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual who could be contacted on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**D.2a**) of the format would identify that individual. The information in the **Document Identification** part of the format (**D.2b**) is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts (**D.2c**, **D.2d**, or **D.2n**) of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: 'The working baseline will only be available to the TAFIM project staff.'" The goal is for the commenter to provide proposed wording that is appropriate for insertion into the document without editing. The D.2 c (5), D.2 d (5), or D.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the document may be provided in the Other Comments field. However, without some degree of specificity these comments may not result in change to the document.

Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 8: DoD Human Computer Interface Style Guide

Version 3.0

30 April 1996

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 8 of the TAFIM, incorporates the following changes from the previous version:

- Chapters 2 and 4 provide more guidance on how to design HCIs, which is the first step in reorientation of the *Style Guide* toward a more process-oriented document.
- Chapters 5 and 6 have had additional material added and the figures updated.
- Minor editorial changes have been made to other chapters.

Future actions with regard to this volume include continuing its evolution toward a "how to design" document, updating the design guidance where needed, exploring methods for compliance, and assessing the impact of the release of Windows95 on the contents of the *Style Guide*. In addition, this volume will be adapted as necessary to reflect the impact of the policy documents and decisions listed above.

A NOTE ON VERSION NUMBERING

The *DoD HCI Style Guide* went through a number of revisions prior to its inclusion in the TAFIM, and as a result has followed a distinctive version numbering scheme outside of the TAFIM system. Version 2.0 of the TAFIM included Version 3.0 of the *Style Guide*. Version 3.0 of the TAFIM includes a version of the *Style Guide* that would have been Version 3.1, but this volume has been designated Version 3.0 of Volume 8 for harmony with the rest of the TAFIM.

A version numbering scheme approved by the Architecture Methodology Working Group (AMWG) will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.

- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version 3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.

- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.

- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the Defense Information Systems Agency (DISA) Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. DISA is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

This page intentionally left blank.

TAFIM Document Configuration Management Page

The latest **authorized versions of the TAFIM** volumes are as follows:

Volume 1:	Overview	3.0	30 April 1996
Volume 2:	Technical Reference Model	3.0	30 April 1996
Volume 3:	Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4:	DoD SBA Planning Guide	3.0	30 April 1996
Volume 5:	Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6:	DoD Goal Security Architecture	3.0	30 April 1996
Volume 7:	Adopted Information Technology Standards	3.0	30 April 1996
Volume 8:	HCI Style Guide	3.0	30 April 1996

Other working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

CONTENTS

FIGURES

1.0 INTRODUCTION

1.1 BACKGROUND

The proliferation of computer technology has resulted in the development of an extensive variety of computer-based systems and the implementation on these systems of varying Human-Computer Interface (HCI) styles. To accommodate the continued growth in computer-based systems, minimize HCI diversity, and improve system performance and reliability, the United States (U.S.) Department of Defense (DoD) is continuing to adopt software development standards. The proliferation of new systems and technology in DoD has also made it necessary to continue efforts to develop and provide guidelines for information display and manipulation.

Computer-based system performance and reliability are products of the performance and reliability of individual components. Computer-based system components include hardware, software, and any user involved in the operation, maintenance, or utilization of the system. Of these components, the user is the most important as well as the most difficult to predict. Thus, a key factor of a high performance, high reliability system is an easy-to-use, effective design of the interface between the user, the hardware, and the software.

One contributor to an easy-to-use, effective HCI is standardization. HCI standardization begins with the selection of an accepted Graphical User Interface (GUI), which in turn provides a standard Application Programming Interface (API) and style approach. Traditionally, the GUI has been determined by the software source selected, such as commercial-off-the-shelf (COTS) software, government-off-the-shelf (GOTS) software, or proprietary software applications. The emerging uniform application program interface (UAPI) technology may free the designer from some of this dependence on the software and hardware platform for the interface "look and feel" (see Section 2.0). The variability of users' needs and differing interpretations of GUI style result in the lack of a common approach and the creation of dissimilar HCIs among systems and applications developed by independent organizations. Adding to the problems in standardization is the fact that the commercial GUI styles do not address issues critical to some DoD organizations, such as geospatial systems, map interface controls, acronym standards, security, and symbol shape standardization.

Standardizing the HCI across application software developed within the DoD community is a two-step process. The first step is to define and document the functional goals, objectives, and requirements of the HCI. The second is for the DoD system and application designers to implement HCI standardization.

1.2 PURPOSE

The purpose of this *DoD HCI Style Guide* (or the *Style Guide)* is to provide a common framework for HCI design and implementation. Through this framework, the long-term functional goals, objectives, and requirements of the HCI will be defined and documented.

Interface implementation options will be standardized, enabling all DoD applications to appear and operate in a reasonably consistent manner.

Specifying appearance, operation, and behavior of DoD software applications will support the following operational objectives:

- **Higher productivity** - People will accept and use what is easy to understand if it aids them in accomplishing their assigned tasks with minimal confusion or frustration.
- **Less training time** - Standard training can be given once for all applications, rather than requiring users be trained when transferring to new systems or new training be created for each new or changed application.
- **Reduced development time** - It will no longer be necessary to design a complete HCI for each system component, because previously developed *Style Guide*-compliant software will be available. The basic appearance and behavior of the interface will be specified by combining and tailoring the commercial GUI style with guidelines in this *Style Guide*. The specific look and feel of each DoD organization's applications software will be detailed in domain-level style guides. These details will limit HCI diversity and further support for the reduction of HCI development time.

1.3 COMPLIANCE

The *DoD HCI Style Guide* has been developed as a guideline document presenting recommendations for good interface design. The *Style Guide* is not intended to be strictly a compliance document; however, it does represent DoD policy concerning HCI design. The interface developer is expected to use the selected commercial GUI style guide, this *Style Guide*, and the appropriate domain-level style guide along with the input of human factors specialists to create the HCI.

The domain-level style guide is the compliance document and may be supplemented by a system-level style guide created as an appendix to the domain-level document. The commercial GUI style guide and this *Style Guide* are expected to be followed in order to maintain consistency and good design principles within DoD. The use of the word "shall" has been eliminated from this document to remove possible conflict of design principles presented with domain-level compliance requirements.

1.4 HUMAN-COMPUTER INTERFACE (HCI)

A user is an integral part of a system. The user-machine interface encompasses interactions between the user and the system, including controls, displays, environmental concerns (e.g., lighting, noise), workspace layout, procedures, and documentation. Design of these elements has a major impact on manpower, personnel selection, training, logistics, safety, and human performance, all of which are elements of concern within DoD systems. HCI addresses the user interface as applied to computer-based systems. HCI encompasses the look and feel of the interface, physical interaction devices, graphical interaction objects, alternate interactions (i.e., voice, touch screen, pen), environmental factors, and any other human-computer interactive methodology. HCI design guidelines in the form of the *Style Guide* provide three major benefits:

- First, the *Style Guide* along with commercial GUI style guides are resources from which designers may draw to aid in developing usable display screens and interactive procedures. This is especially important because the rapid pace of knowledge acquisition impacts human performance and computer systems, and because the GUI has emerged as the dominant architecture for the HCI.

- Second, the guidelines provide a common approach that supports consistency of design a fundamental principle of human factors engineering design.

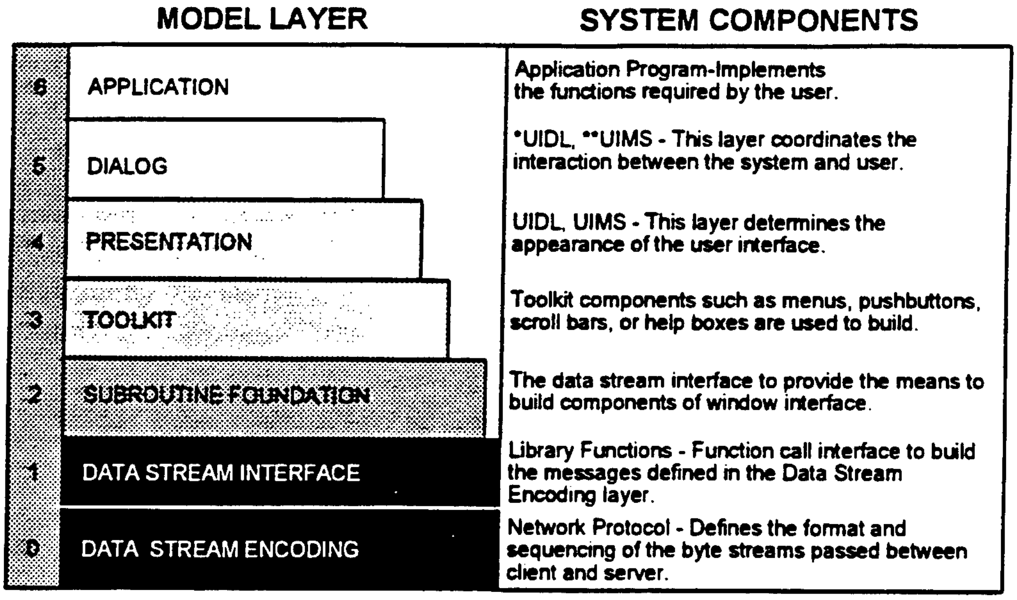
- Third, the guidelines will allow for a broader range of personnel selection criteria, and will reduce training and possibly manpower requirements for all systems.

1.5 SCOPE

Two factors influence the applicability of the *Style Guide* to DoD computer-based systems: the software architecture being used and the functional requirements of the specific system. This *Style Guide* addresses functional requirements and operations that are intended by DoD to be consistent across the entire interface design. The *Style Guide* emphasis is on HCI considerations for features and functions applicable to DoD systems and applications. Such features and functions include system start-up, security issues, and map graphics.

The *Style Guide* has been developed to address design considerations germane to the DoD environment. The guidelines are generic enough to apply to almost any GUI and, to a lesser extent, to text-based interfaces. The system developer needs to be aware that using a software architecture other than those mandated for use within DoD will limit portability to and reusability by other systems within DoD.

Guidelines are presented for application development within layers 0 through 5 of the National Institute of Standards and Technology (NIST) reference model. Figure 1-1 presents a summary of the NIST reference model. For layers 0 through 2, applications should adhere to the X Window processing standards in Federal Information Processing Standard (FIPS) 158. Layer 3 defines the toolkit standards that support the window management API. Layers 4 and 5 define the look and feel of the GUI. Standards for the upper layers are currently under development by IEEE P1201 committees and may eventually be incorporated into this *Style Guide*. These guidelines define how user interface services are to be provided within the DoD technical reference model (TRM). The TRM, shown in Figure 1-2, defines the set of services to be provided by the application platform and the associated profile of standards for implementing the services.



MODEL LAYER		SYSTEM COMPONENTS
6	APPLICATION	Application Program-Implements the functions required by the user.
5	DIALOG	*UIDL, **UIMS - This layer coordinates the interaction between the system and user.
4	PRESENTATION	UIDL, UIMS - This layer determines the appearance of the user interface.
3	TOOLKIT	Toolkit components such as menus, pushbuttons, scroll bars, or help boxes are used to build.
2	SUBROUTINE FOUNDATION	The data stream interface to provide the means to build components of window interface.
1	DATA STREAM INTERFACE	Library Functions - Function call interface to build the messages defined in the Data Stream Encoding layer.
0	DATA STREAM ENCODING	Network Protocol - Defines the format and sequencing of the byte streams passed between client and server.

* User Interface Definition Language (UIDL)
** User Interface Management System (UIMS)

Figure 1-1. NIST User Interface System Reference Model

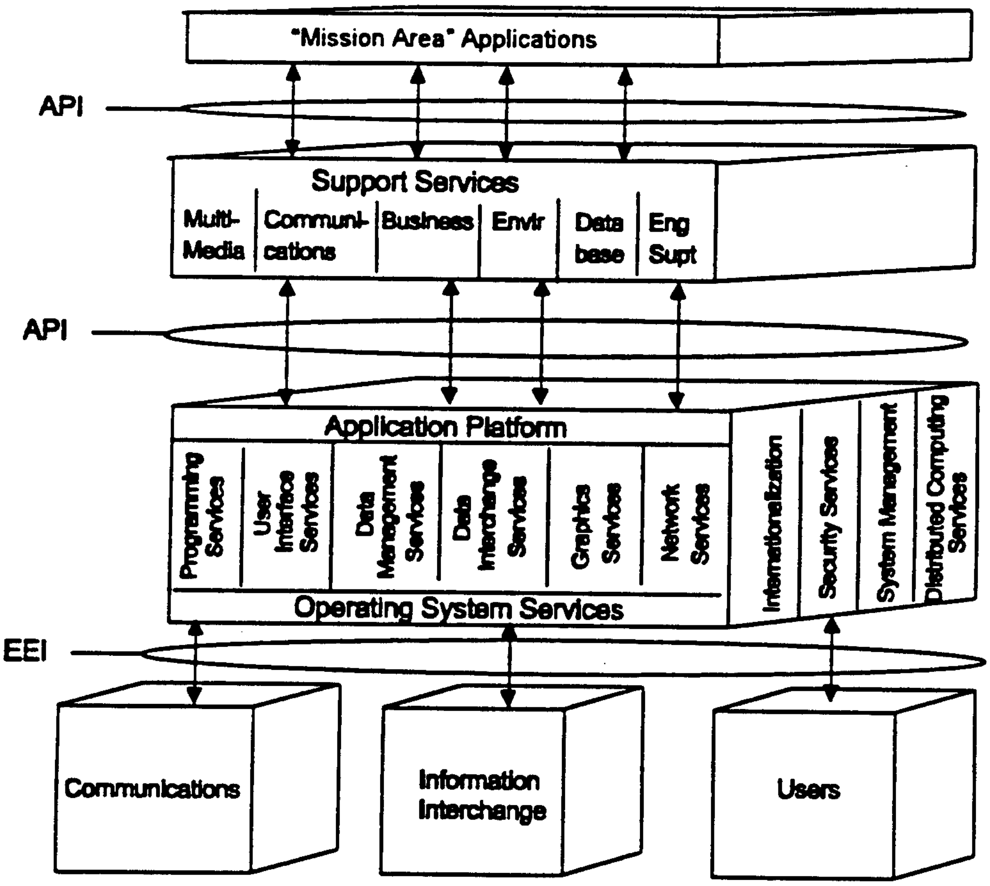


Figure 1-2. DoD Technical Reference Model

1.6 INTENDED AUDIENCE

The target audience for this *Style Guide* includes DoD military and civilian personnel along with contractors representing those who determine system requirements, program managers, system managers, software developers, and application HCI designers. Ideally, these individuals should be knowledgeable of the characteristics of the intended user population and the tasks these users must perform. In addition, the users of this *Style Guide* should have some knowledge of human-performance considerations. A secondary audience includes users and software maintainers who are interested in the general design of the interface, who wish to provide feedback concerning modifications and improvements to the *Style Guide*, or who wish to assess the usability of fielded systems or applications in terms of their compliance with *Style Guide* content.

There are two basic environments within DoD that the *Style Guide* addresses: the operational and the business. The operational environment includes both strategic and tactical systems, though not necessarily mission-critical weapons systems. The business environment includes systems used in military and civilian office environments. Systems from within the operational environment are moving towards the use of UNIX, Open Software Foundation (OSF)/Motif, and Open Look for the user interface, whereas the business environment tends to use Microsoft Windows, OS/2 Presentation Manager, and the Apple Macintosh interface styles. A critical difference between the two environments involves the degree of customization that is recommended. In the business environment, with its more stable user community, individual customization is more acceptable. In the operational community, with its higher turnover of users, multiple users, and need for over-the-shoulder viewing, individual customization can have a negative impact on human performance and should be used cautiously.

1.7 DESIGN GOALS

DoD application development should:

- First, identify and be familiar with the functions and tasks to be performed by the system and the operational environment. This allows development of an understanding of the overall system dynamics.

- Second, complete an analysis of the capabilities and limitations of system users. A task trade-off analysis between the user and the application is recommended. This allows development of an understanding of which tasks are best performed by the human and which are best performed by the hardware and software. In addition, this understanding provides the groundwork for task and interface design to ensure that the user can successfully perform the required tasks.

- Finally, apply a consistent set of rules for designing the interface. The rules for the design of the HCI include, but are not limited to, the following:

  - Design the applications to meet specific user requirements. Above all, provide the functionality to meet those requirements.

- Ensure that all applications are consistent with the interface guidelines specified in the appropriate commercial GUI style guide, in this *Style Guide*, in the domain-level style guide, and in the system-level specifications.
- Ensure that an application's HCI provides rapid access to all of its functions. To ensure this, avoid unnecessary menus and long selection lists that force users to "page" through all entries.
- Ensure that the application is flexible. For example, provide multiple methods to access a function (e.g., direct command line entry, menus, tree diagrams, mnemonics, and keyboard accelerators).
- Require explicit action to perform any act that could result in irreversible negative consequences, and provide users with options (e.g., quit without saving).
- Give users a choice of input devices (keyboard or pointing device) for scrolling, map manipulation, and invoking or terminating an application. The keyboard and pointing device should be interchangeable where appropriate to the action being performed.
- Ensure that an application user interface does not depend on color to communicate with the user. Color should add substance to the interface, not dominate it.

1.8 ASSUMPTIONS

In writing this *Style Guide*, the following assumptions were made:

- The user will interface with information from external systems, COTS software, and GOTS applications.
- The application design requirements specified in this *Style Guide* will be supported by standard DoD civilian computer environments, and tactical or strategic computer environments. The DoD HCI will be implemented on a variety of computer architectures. Computer systems will be equipped with diverse capabilities, such as monochrome versus color monitors and varying amounts of random access memory.
- The *Style Guide* will not address all elements of the human-machine interface. The focus of this document is on the HCI within DoD.
- A system will be composed of a set of applications and will meet the operational needs of users through the integration of multiple applications from a variety of sources (e.g., COTS, GOTS).

1.9 *STYLE GUIDE* ORGANIZATION

Section 2.0 of the *Style Guide* describes the interface style and design issues that must be addressed by software developers within DoD. This section also addresses the concept of application portability between platforms and between GUI styles.

Section 3.0 describes hardware considerations, with focus on input/output devices and their alternatives. This section includes issues related to the Computer/Electronic Accommodation Program (CAP). A subsection on special displays is also included.

Sections 4.0 through 10.0 contain HCI guidelines for the designer. General subjects covered include screen design, windows, menu design, object orientation, common features, text, and graphics. Each section is divided into specific subject areas and includes examples of the stated design guidelines.

Sections 11.0 through 13.0 cover application design guidelines. The topics include decision aids, query, and embedded training. The selected applications represent focus areas of DoD applications and subjects that have generated questions and comments from system developers.

Section 14.0 covers emerging technologies, with initial information on guideline considerations for new areas. This section addresses topics that may become additional sections in later versions or may be added to existing sections.

Appendix A describes objective security interface requirements, using the *DIA Style Guide* and DDS-2600-6215-89 as baselines.

Appendix B, the glossary, defines frequently used terms pertaining to the HCI and GUI style guidelines.

Appendix C, references, is supplemented by direct references at the end of each section, providing a means to determine the original source of a specific guideline. Appendix C demonstrates the overall review undertaken to provide a baseline for this document.

Addenda will describe specific interface requirements of various organizations served by this *Style Guide*. This version of the *Style Guide* includes by reference "User Interface Specifications For The Joint Maritime Command Information System (JMCIS), Version 1.3" as Addendum 1. Additional addenda will be added as required.

1.10 BASELINE

The users of this *Style Guide* should seek out the following references for use in interface development:

- *Air Force Intelligence Data Handling System Style Guide* (U.S. Air Force 1990) establishes HCI guidelines for applications developed for Air Force Intelligence analysts and users.
- Blattner, M. M., and R. B. Dannenberg, Multimedia Interface Design, ACM Press, 1992.
- The *Defense Intelligence Agency (DIA) Standard User Interface Style Guide for Compartmented Mode Workstations* (DIA 1983, henceforth called the *DIA Style Guide*) and *Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines, Rev. 1* (Final) (DIA 1991, henceforth called DDS-2600-6215-91). These documents address

the security portion of the HCI and are intended for designers of applications for compartmented mode workstations (CMW). They outline security-related interface requirements for workstations operating in the system high or compartmented mode.

- The *Department of Defense Intelligence Information Systems (DODIIS) Style Guide*, (DODIIS 1991a) from which Version 1.0 of this *Style Guide* was adapted.

- *DoD Human-Computer Interface Style Guide*, Versions 1.0, 2.0, and 3.0 (1992a, 1992b, 1993), which provide a framework focused on designing the user-computer interface to enhance user performance.

- FIPS 158-1, "User Interface Component of Applications Portability Profile" (NIST 1993), which mandates the use of the X Window protocol, X library, and X toolkit intrinsics.

- Galitz, W. O., *User-Interface Screen Design*, QED Information Sciences, 1993.

- *Human Engineering Design Criteria for Military Systems, Equipment and Facilities*, MIL-STD-1472D (DoD 1989b) and *Human Engineering Guidelines for Management Information Systems*, DOD-HDBK-761A (DoD 1989c), both of which are human factors standards DoD currently uses.

- *Human Factors Guidelines for the Army Tactical Command and Control System Soldier-Machine Interface*, Versions 1.0 and 2.0 (Avery et al. 1990 and 1992), which provide a set of overarching guidelines focused on designing the user-computer interface to enhance user performance.

- "Institute of Electrical and Electronics Engineers (IEEE) Recommended Practices for Graphical User Interface Drivability," Draft 2 (IEEE 1993b, henceforth called IEEE P1201.2). When adopted, this document will standardize those HCI elements and characteristics that must be consistent to facilitate users switching from one look and feel or application to another.

- Kobara, S., *Visual Design with OSF/Motif*, Addison-Wesley Publishing Company, 1991.

- NIST User Interface System Reference Model, as found in *Volume 2: The Technical Reference Model and Standards Profile Summary*, Version 2.0, (Defense Information Systems Agency [DISA], June 1994), has been adopted as the baseline for this document.

- NIST Special Report 500-187, *Application Portability Profile (APP): The U.S. Government's Open Systems Environment (OSE) Profile OSE/1*, Version 1.0, May 1991.

- North Atlantic Treaty Organization (NATO) Standardization Agreement 2019, *Military Symbols for Land Based Systems* (NATO 1990); Army Field Manual 101-5-1, *Operational Terms and Symbols* (U.S. Army 1985b); and "DIA Standard Military Graphics Symbols Manual" (DIA 1990 Draft), which standardize map graphics symbols.

- The *Open Look Graphical User Interface Application Style Guidelines* (Sun Microsystems, Inc., 1990); *Open Software Foundation (OSF)/Motif™ Style Guide*, Revision 1.2 (OSF 1992); *The Windows™ Interface: An Application Design Guide*, Microsoft Press, 1992; *MacIntosh Human Interface Guidelines*, Apple Computer, Inc., 1992, which describe the major X Window GUIs; and MIL-STD-2525, *Common Warfighting Symbology*, Version 1 (DoD 1994).

- *User Interface Specifications For The Joint Maritime Command Information System (JMCIS)*, Version 1.3 (Fernandes 1993), which defines a common look and feel for Navy command and control systems.

This *Style Guide* draws from the aforementioned documents. The intent is to establish style objectives and guidelines to which all members of the DoD community can transition.

This page intentionally left blank.

2.0 INTERFACE STYLE

Given the direction of technology development for the HCI, a long-term goal of DoD has been the implementation of a more common, standardized interface style. FIPS 158 (NIST 1990b) was originally implemented to provide guidance to DoD system designer/developers to encourage standardization of the "look and feel" (i.e., interface style) through the use of a common windowing architecture. FIPS 158 was also interpreted to mean that the developer should use either Open Look or Motif as an interface standard to ensure compliance with this standardization. This focus led to the development of earlier versions of the *DoD HCI Style Guide*, which encouraged the use of these styles. Due to changing technology, DoD is now more broadly interpreting the implications of FIPS 158 on interface style. The reasons for this broad interpretation include:

- The emerging capability of other interface styles, such as Apple MacIntosh and Microsoft Windows and others, to operate on top of X Window

- The concern for providing guidance for both the operational (e.g., tactical) and business environments within DoD

- The emergence of the UAPI environment tools that allow portability from one computer platform to another.

While X Window provides the underlying technology through which HCI interfaces of different types will achieve standardization and portability, FIPS 158 is now being interpreted to allow for the use of any of the standard interface styles. These interface styles consist of Open Look, Motif, MacIntosh, Microsoft Windows, and OS/2 Presentation Manager. This broader interpretation, while providing greater flexibility to interface designers, also increases the potential for reduced consistency/commonality. Therefore, the need for style guides in general, and this *Style Guide* specifically, becomes all the more important. Use of these style guides will ensure that HCIs are developed in accordance with sound principles of interface design and that consistency and commonality are encouraged. The objective of Section 2.0 is to provide the reader with an understanding of both how the *Style Guide* should now be used in interface design and system development, and how the emerging UAPI tools impact system design and the *Style Guide*.

2.1 STYLE GUIDES

Good software design requires selecting and using standard practices for various aspects of an application or system design. This helps ensure consistency of appearance and behavior among applications within a system and develops designs to enhance human performance. A number of documents are available that can help the system designer provide guidance to the HCI designer, including standards (e.g., MIL-STD-1472D, DoD 1989b and IEEE P1295, IEEE 1993c), handbooks (e.g., MIL-HDBK-761A, DoD 1989c), and style guides. Of these documents, the style guides may be the most helpful to HCI designers. Several categories of style guides are available (see Figure 2-1, Style Guide Hierarchy) to a system designer/developer once the specific GUI style has been selected. The style guide hierarchy begins with the commercial style guides and is refined by the *DoD HCI Style Guide,* with specific style decisions given in the domain-level style guide. The detail proceeds from the general "look" of the interface through to specific functionality. Each category of style guide is discussed in Paragraphs 2.1.1 through 2.1.4.

2.1.1 Commercial Style Guides

The style guides provided by major software vendors and consortia cover horizontal aspects of effective design, or those aspects applicable to the widest breadth of systems, applications, and domains. Commercial style guides provide standard design practices for specific development environments, such as Motif or Windows. The commercial style guide will provide a broad understanding of how the system will look and, to a certain degree feel, based on the software architecture underpinning the system.

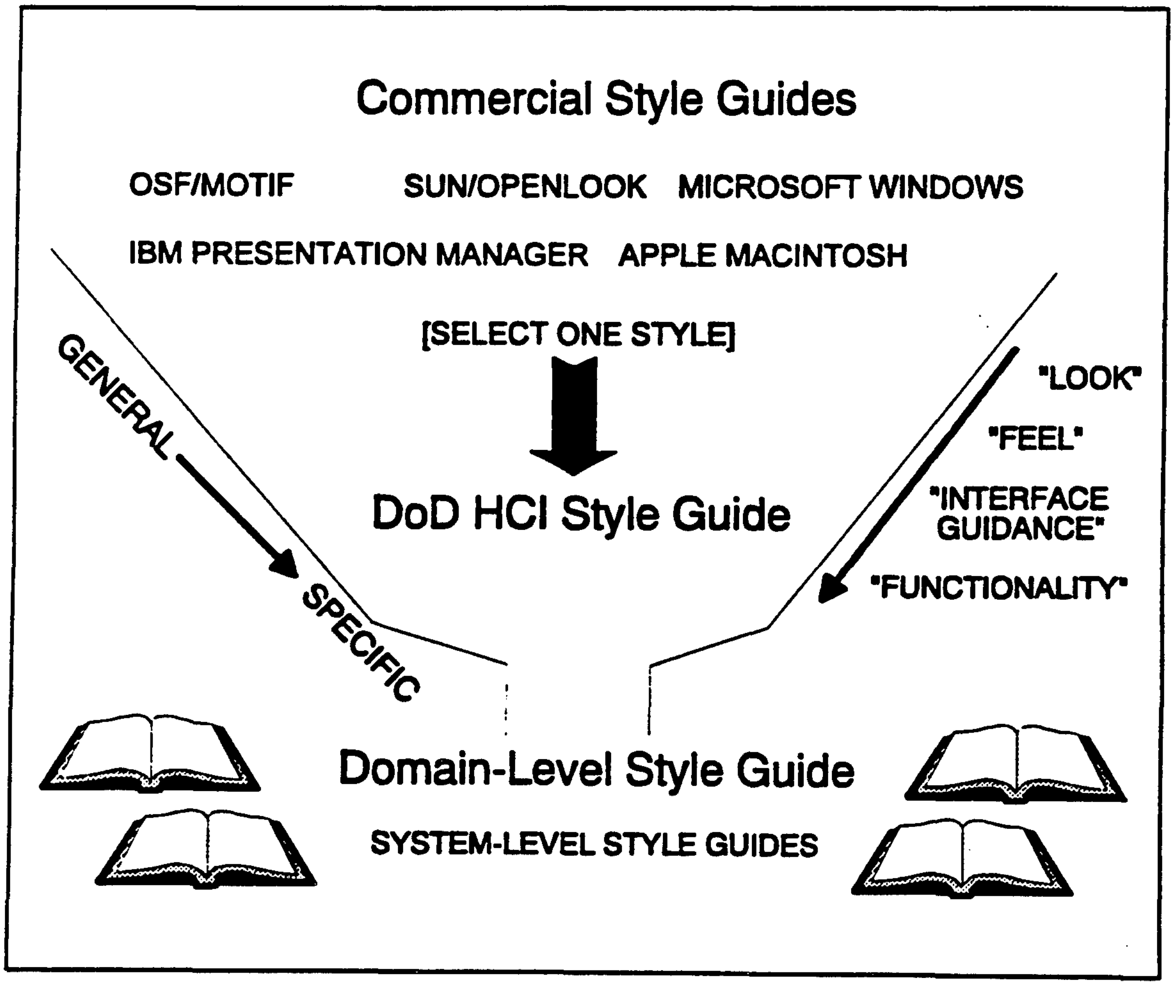


Figure 2-1. Style Guide Hierarchy

Commercial style guides do not necessarily address human performance or military system considerations, but rather more general software behavior. Commercial style guides will provide general guidance that allows a system to deliver a consistent style if a single GUI, such as Motif or Windows, is used. However, the specific style defined by one GUI may differ from that for another GUI, so inconsistencies arise if different GUIs are available on a single platform or workstation.

The commercial style guides contain numerous stylistic differences due to different approaches taken by each vendor. These differences can be grouped into the following broad categories:

- **Terminology** - differences in names assigned to, and descriptions of, functions and features. Commercial style guides use substantially different terms to describe the functions and features associated with their respective GUIs. The main distinction is that different terms and descriptive phrases are used to define and describe equivalent or similar functions and features. However, in some instances, the same term is used to refer to different, unrelated functions or features. An example of using different terminology to describe similar functions is: Motif uses the term "radio button" and Windows uses "option button." Both terms refer to lists of selections for which only one choice can be made.

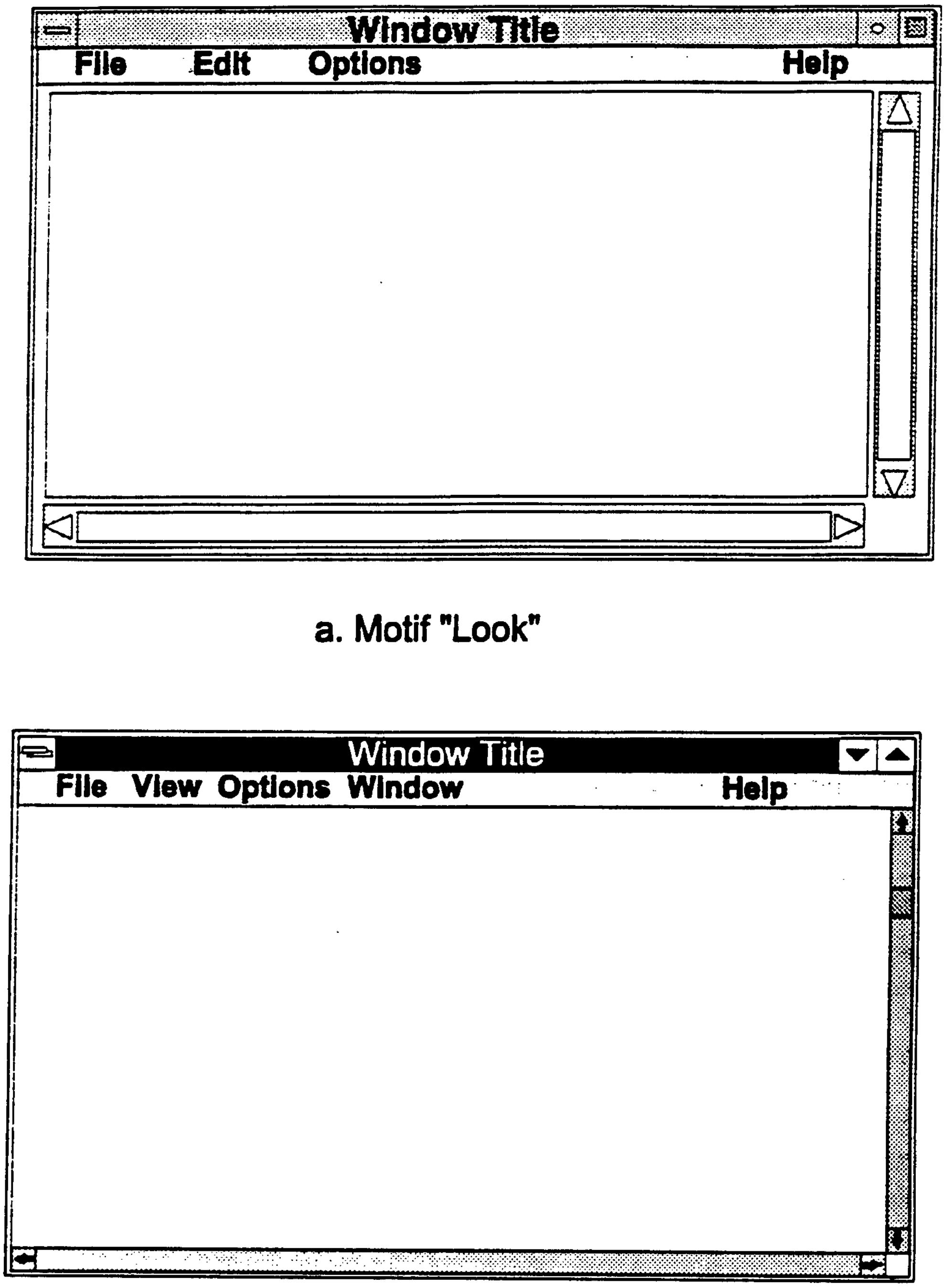
- **Look** - differences in the appearance of displays based upon different styles. The concept of look can be illustrated by comparing the graphic representations (see Figures 2-2a through 2-2d) of each major style.

- **Feel** - differences in the actions a user takes to interact with an application. For example, the differences in the feel of Motif and Windows interfaces are illustrated by the application of keyboard special-purpose keys, mnemonics, and accelerators; and by the use of some special-purpose controls. Both Motif and Windows support keyboard input, but there is very little consistency between the two GUIs in defining special-purpose keys.

2.1.2 The *DoD HCI Style Guide*

The *DoD HCI Style Guide* provides an additional source of interface design input along with commercial style guides that can be used by a system developer/designer. The *Style Guide* addresses common user interface design issues, contains guidance derived from research on human performance, and provides a focus on elements applicable to DoD systems.

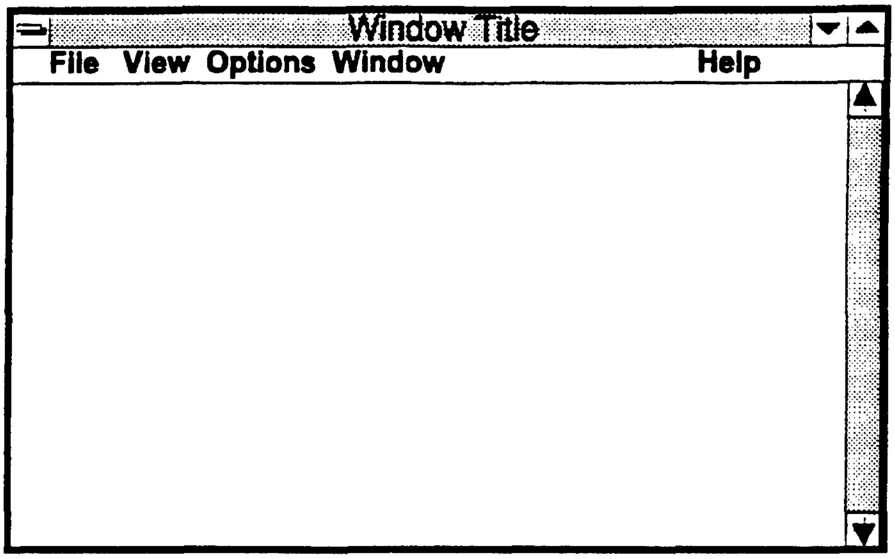
The *Style Guide* promotes consistency by providing generic guidelines that can be applied across the multiple GUIs in use within the DoD environment today. The *Style Guide* provides additional performance-based guidelines for use in designing GUI elements defined within the commercial style guides (e.g., menu and function names, accelerator keys, and mnemonics). The *Style Guide* addresses functional areas applicable to DoD systems not addressed within the commercial style guides (e.g., security classification markings, tactical color codes) and includes appendixes that identify domain-level style guides currently available for the services and other DoD organizations.



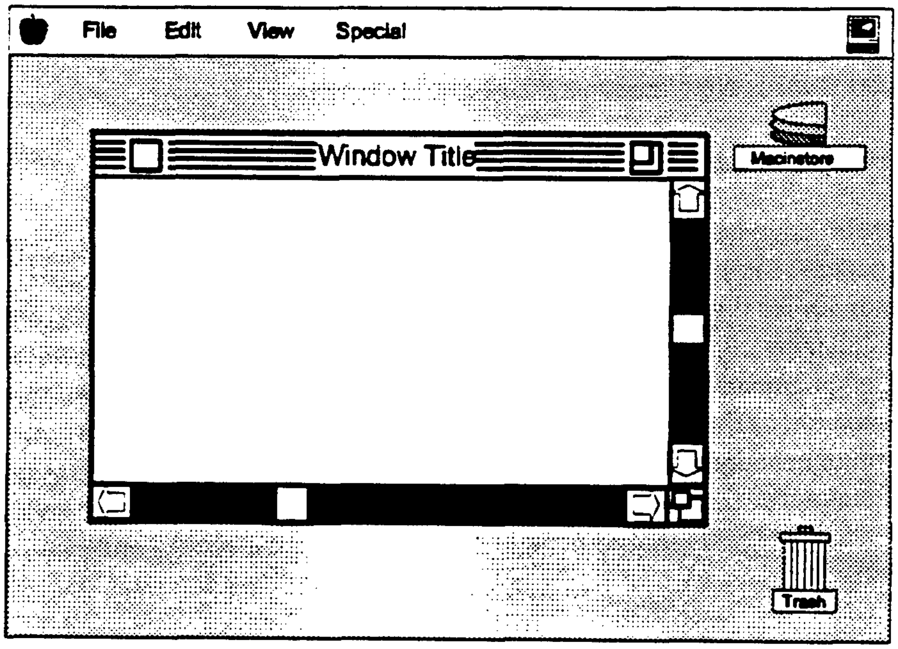
a. Motif "Look"

b. Windows "Look"

Figures 2-2 a and b. Different Window "Looks"



c. OS/2 "Look"



d. Macintosh "Look"

Figures 2-2 c and d. Different Window "Looks"

Beginning with Version 3.1, the *Style Guide* will evolve into a new format focusing on "How To" guidance for combining commercial style guide information with current standards and DoD HCI design considerations. The intent of the new format is to provide complementary information to that available in domain-level style guides, which are intended to provide more detailed "What To Do" specifications. The changes within the *Style Guide* will be staged over the next several revisions due to budget and time constraints. When sections are converted to the How To format, the revised sections will provide specific examples and How To guidance including examples of "good" and "bad" design where appropriate. The successive versions of the *Style Guide* will be reviewed to continue eliminating duplication between it and commercial style guides.

2.1.3 Domain-Level Style Guides

Domain-level style guides provide detailed guidance that addresses the requirements of a particular domain (e.g., Command, Control, Communications, Computers, and Intelligence [C4I] and space) as defined by a DoD organization (e.g., joint, individual service, or agency). Domain-level style guides reflect the consensus of the organization on the look and feel they want to provide in their systems. Over time, it is expected that DoD organizations will develop and publish domain-level style guides for directing the HCI design efforts for their systems. An example of a domain-level style guide is the *User Interface Specifications For The Joint Maritime Command Information System (JMCIS), Version 1.3* (Fernandes 1993), which defines a common look and feel for Navy command and control systems.

2.1.4 System-Level Style Guides

A system-level style guide, when developed, is used to address system issues and to provide design rules for that specific system. When system-level style guides are used, the look and feel provided in the domain-level style guide is to be maintained. The system-level style guide will provide the "special" tailoring of the commercial, DoD, and domain-level style guides and will include explicit design guidance and rules for the system, as well as document design decisions made during the creation of the user interface. Other style guides may be available from commercial or government sources for a specific application being developed. The system developer should make these documents available to the HCI developer, identify them as reference documents, and call them out in the application-specific technical specification and design documentation.

2.2 SYSTEM-LEVEL USER INTERFACE DESIGN DECISIONS

2.2.1 Selecting a User Interface Style

The first design decision made for a new system should be the primary style under which the system will be fielded, usually driven by the selection of the hardware and software architecture. The commercial styles most frequently used within DoD include OSF/Motif, Sun/Open Look, Microsoft Windows, IBM Presentation Manager, and Apple MacIntosh. However, the preferred style for all DoD tactical applications is OSF/Motif. It should also be noted that the use of Open

Look is discouraged on new DoD systems due to its increasing convergence with Motif and its decreasing use in the general marketplace.

Because the DoD software architecture allows systems to use various commercial styles, the *Style Guide* was developed to address design considerations germane to most style environments. However, regardless of the interface, applications should adhere to the X Window processing standards in FIPS 158.

2.2.2 Deciding on a System-Level Style Guide

When required, system-level style guides, with system here defined as a family of applications, represent the tailoring of vendor, DoD, and domain-level guides to meet the special needs of the system being developed. The goal of the system-level style guide is to ensure the development of a standardized, coherent, and usable HCI. A system developer should:

- Select a domain-level style guide, if one is available for the domain and GUI (Assume the domain style guide has evolved from the *Style Guide*).
- Define a system-specific appendix to the domain style guide, if there are system-unique requirements not addressed in that style guide.
- Develop a separate system-level style guide only if an appropriate domain-level document is not available. The system-level style guide should use the relevant commercial style guide and the *Style Guide* as starting points for its content, with tailoring as needed to meet system requirements.

2.2.3 HCI Design Process

The system designer/developer should make available all appropriate levels and types of style guides for use in designing the HCI. Figure 2-3 illustrates the process by which a design is evolved from the different types of style guides, in essence moving from the general to the specific. The system concept is then derived from the interpretation of requirements within the guidelines of the standards, style guides, and functionality. While developing the system-level design guidelines and rules, the design should be prototyped as a way to explore and refine concepts with representatives of the user population. This concept exploration will usually help clarify the system requirements and identify aspects of the design or interface style that require special interpretation of the domain-level style guide and/or the creation of a system-level style guide.

2.2.4 Migration Strategy

The goal of the DoD migration strategy is to transition existing information-processing systems to a single HCI within an open system architecture. Current DoD policy calls for the HCI to be based on the X Window system in order to provide interoperability among systems. The intent of a DoD migration strategy is to define a generic process that can be applied by all of its systems to achieving this goal.

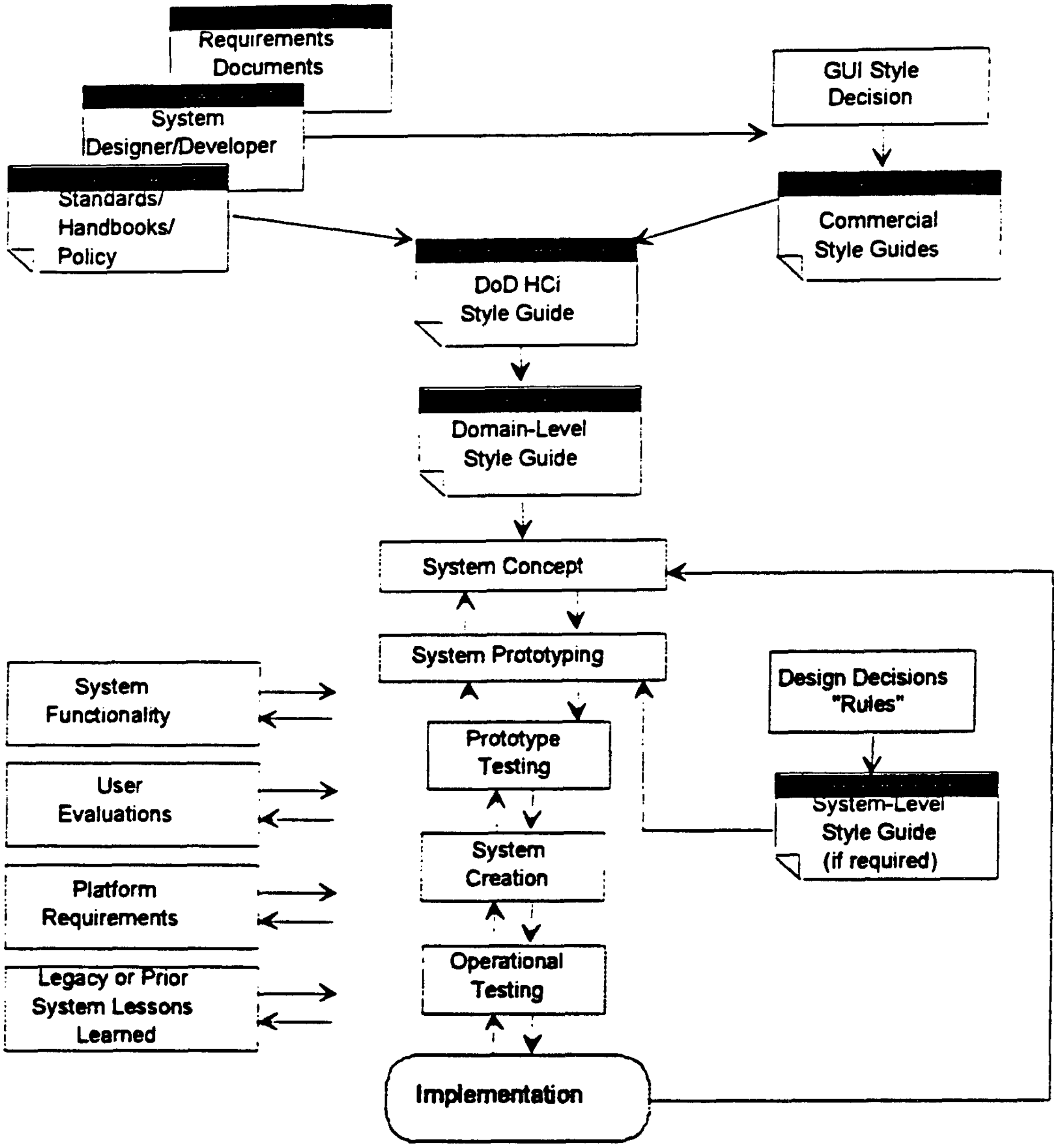


Figure 2-3. HCI Design Process

DoD migration strategy is conceptualized as a process with short-term, intermediate, and long-term objectives. The short-term objective is to transition existing systems from their current user interface (e.g., one that is character-based) to one that is GUI-based. Because this transition allows systems to implement GUIs with different styles, the intermediate objective of the migration process is to maximize the common user interface features available within these different styles. The creation of domain-level style guides as compliance documents is a step in the transition to a common interface style and a standard HCI. Although providing a single HCI based on an open system architecture represents a long-term goal, the transition process toward a common user interface style is one that can and should be undertaken by all DoD systems.

2.2.5 Portability Across Hardware Platforms

A critical concern for HCI developers within DoD is how to build an interface on one type of platform and then easily replicate that interface on diverse hardware platforms, either retaining the original interface style or taking on the style native to the new platform while maintaining standardization. A new, emerging technology that may have an impact on this concern and the HCI design process is UAPI. This technology enables the porting of HCI applications from one platform to another and is described in more detail in Subsection 2.4.

2.2.6 Integration of HCI Environments

The integration of business, tactical, finance, personnel, and all other DoD computer interface environments to common HCI principles is a long term goal of DISA. Each of these environments shares common interface issues while at the same time each represents unique interface approaches. The interface guidelines presented in *Style Guide* Sections 3.0 through 14.0 are intended to address common interface issues. The principles of good interface design should be applied to all HCIs used within DoD. The goal of good interface design is to provide the user with the tools needed to complete the required tasks with the greatest ease and effectiveness.

The general difference between the various environments can be described in terms of the usual software within the environment. The business environment is characterized by the use of COTS software as the prime source of application software. The extensive use of COTS software reduces the ability of the system developer to affect the HCI design for the application. The tactical environment has the highest degree of custom-developed software applications, and the result has been the greatest diversity of interface styles and designs. The financial environment carries the legacy of mainframe applications that are oriented to command-line and text-based interfaces. The personnel and logistics applications have the largest databases (other than geographic data) of any of the DoD environments. The maintenance of the database input/output is the focus of these interfaces. The specialized interfaces, such as those used in real time weapon system application, have interface requirements that are beyond the scope of the *Style Guide*. The creation of domain-level style guides is especially important to those systems not completely covered in the *Style Guide*. The general principles given in this document apply to all interfaces, but some specialized areas require separate consideration.

2.3 USING THE *STYLE GUIDE* TO SOLVE USER INTERFACE DESIGN PROBLEMS

Integrating all DoD HCI environments (i.e., business, tactical, finance, personnel) to common HCI principles is a long-term goal of DoD. Each of these environments shares common interface problems, while at the same time each has unique interface requirements. The following paragraphs address the *Style Guide* approach to the common problems and provide guidance for applying the principles so that users are provided with the tools needed to complete the required tasks with the greatest ease and effectiveness.

2.3.1 Selecting a User Interface Style

a. PROBLEM: Many commercial applications in office environments use Microsoft Windows. Additionally, an increasing number of commercial applications are available with either the Motif GUI or with the Apple/MacIntosh GUI.

b. RECOMMENDATION: A single GUI should be selected for use within a work group. Choices include Microsoft Windows, Apple/MacIntosh, OS/2 Presentation Manager, or Motif.

2.3.2 Redesigning the HCI to Improve Usability

a. PROBLEM: The software was not designed to do the task(s) to which it is currently applied. It follows that the labels, headings, and indicators are not consistent with the user requirements.

RECOMMENDATION: The interface should be redesigned as soon as possible, because continued use of an inappropriate interface will reduce productivity and lower morale. Sections 6.0 and 9.0 apply to this problem.

b. PROBLEM: The software has been designed to mirror a non-automated (i.e., paper) system without elimination of duplicate inputs, and uses input formats that are not optimized for the computer.

RECOMMENDATION: The interface should be redesigned as soon as possible since continued use of an inappropriate interface will reduce productivity and lower morale.

c. PROBLEM: Terminology, jargon, acronyms, capitalization, and abbreviations are not consistent with the users' expectations and common understanding.

RECOMMENDATION: These aspects of the interface can cause critical errors in operation and reduce productivity. The software should be revised or upgraded as soon as possible in these circumstances. Sections 8.0 and 9.0 apply to this problem.

d. PROBLEM: The task sequence within the software is not consistent with the operational tasks the operator/user is required to accomplish using the software. In some cases, the use of a software application may take more time and effort than the corresponding manual system.

RECOMMENDATION: The requirements/specifications for the software should be reviewed and redesign undertaken, if appropriate. Sections 6.0 and 9.0 apply to this problem.

e. PROBLEM: The application extensively uses data available in other applications, but no interoperability or connectivity is supplied. The operator/user spends large time sequences in duplicate data entry.

RECOMMENDATION: The data entry process is error prone and should be minimized where possible. The use of interconnectivity to reduce duplicate data entry is encouraged. Information in Section 9.0 applies to this problem.

f. PROBLEM: The application software employs codes and/or procedures from prior software applications that are difficult to remember but no longer required due to changes in technology.

RECOMMENDATION: The interface should be designed to simplify the users' tasks and take advantage of improved technology. The requirement to use cryptic input codes should be eliminated wherever possible.

g. PROBLEM: The software is very complex and requires extensive operator/ user training to make effective use of its capabilities. The result is that the software is rarely or never used, with subsequent loss of the capability offered by the application.

RECOMMENDATION: The addition of software navigation aids, improved HELP, and possibly on-line tutorials should be considered in cases where complete redesigns are not cost-effective. Sections 6.0 and 8.0 apply to this problem.

2.3.3 HCI Considerations in Selecting Commercial Software

a. PROBLEM: The primary source of application software in a particular domain may be COTS software packages. This may be a problem because the COTS software has a great deal of variability in quality of interface design.

RECOMMENDATION: Evaluation copies of proposed software purchases should be subjected to compliance evaluation based upon the domain-level style guide or, if one is not available, the *Style Guide*. This should occur prior to procurement of multiple copies. The procurement of COTS software should provide for the comparison of applications with parallel functionality (i.e., Word Processor with Word Processor; Spreadsheet with Spreadsheet). Comparison should include user evaluation, HCI evaluation, functionality, and compliance with the appropriate domain-level style guide and/or the *Style Guide*.

2.3.4 HCI Considerations in Developing Custom Software

a. PROBLEM: The acquisition of custom software introduces nonstandard GUIs into the environment. There is also an increase in the diversity of the HCI look and feel due to stovepipe development if more than one custom system is developed.

RECOMMENDATION: The procurement of custom software applications should be required to be in compliance with the applicable domain-level style guide or if one is not available the *Style Guide*. The standard commercial interface style that is used by the domain (environment) that will use the software should be specified for the application unless it is not a GUI. If the interface style normally used is not a GUI, the specification should be directed to an accepted GUI.

2.3.5 HCI Design in Tactical Environments

a. PROBLEM: The tactical environment frequently involves operator/users using the same application on the same hardware in shifts. The consistency of look and feel is increased in importance under these conditions.

RECOMMENDATION: Compliance with the *Style Guide* and appropriate domain style guide should be combined with compliance to system-level specification and style guide (if needed) to establish as much consistency as possible within and between sets of applications available on the system.

b. PROBLEM: Tactical applications frequently use maps as the basic screen background. Map usage is encouraged but presents difficulties in background foreground contrast, clutter, resolution, and system response time.

RECOMMENDATION: These issues must be addressed in HCI design. See Section 10.0 for more information.

c. PROBLEM: The tactical environment frequently has difficulty maintaining the availability of trained operators and circumstantially may require partially trained individuals to operate a given application.

RECOMMENDATION: This problem increases the importance of the HELP system, embedded training, ease of operation, and consistency of the interface. See Sections 8.0, 13.0, and 14.0 for more information.

d. PROBLEM: The tactical environment frequently creates a high stress level on the operator/user during use of the application. The high stress environment makes operators more error-prone in their interaction with the application.

RECOMMENDATION: Careful attention should be given to error management within tactical applications. See Section 8.0 for information on HELP systems and Subsection 9.2 on form filling.

e. PROBLEM: The use of multiple operators on the same hardware and application requires maintaining a consistent interface.

RECOMMENDATION: Although commercial software offers configuration and color choices to the operator, offering the choices is not recommended in cases where multiple operators share the use of the same equipment. See Subsection 4.3 for more information.

f. PROBLEM: The use of color in the tactical environment has preassigned specific meaning.

RECOMMENDATION: The use of color and color combination must be carefully planned and controlled in tactical applications. See Subsection 4.3 for more information on color.

2.3.6 Migration Considerations

a. PROBLEM: The interface is either "command line" or "text based" with the experienced users resisting change and new users requiring extensive training.

RECOMMENDATION: The DoD goal is to convert to GUI as soon as possible. However, in these cases, consideration should be given to allowing access to the original interface as a subset of the HCI to provide a transition for experienced users. Sections 5.0 and 7.0 apply to this problem.

b. PROBLEM: The software is different (not consistent) in look and feel from other applications in the same environment.

RECOMMENDATION: The goal of consistent look and feel within DoD applications should be a factor in determining application upgrades and replacements. Sections 6.0 and 7.0 apply to this problem.

2.3.7 Portability Considerations

a. PROBLEM: The software was not designed for the hardware system on which it is being used and contains inappropriate operator actions or is excessively slow in executing commands.

RECOMMENDATION: The use of one of the methods for transporting software described in Subsection 2.4 should be reviewed along with an investigation into the cost benefit of upgrading the software and hardware.

b. PROBLEM: Individual users employ more than one workstation or share a workstation with other users.

RECOMMENDATION: A personal layer system (see Subsection 14.1) should be created.

2.4 UNIFORM APPLICATION PROGRAM INTERFACE (UAPI)

2.4.1 Introduction

Application program portability from one computer platform to another is an OSE goal for DoD. The advent of GUI technology provided flexibility for HCI design and opened new options, while introducing complications for cross-platform compatibility of application programs. As a result, there is a heightened need for ensuring that consistent GUI software design is planned deliberately and appropriately.

The fact that designers have chosen user interface styles that are compliant with the *Style Guide* to ensure compatibility with X Window has contributed to possible portability of applications. Thus, style restrictions become less of an issue. The FIPS 158 (NIST 1990b) acceptability of alternate development environments broadens as GUI application development environments begin to allow for planned switching from one HCI style to another. This broader view is enhanced as applications are transported from one host platform to another.

These developments have special significance within DoD, given the diverse needs of its two basic environments -- operational and business. The operational environment has a greater need for HCI application interfaces that appear and behave the same, regardless of the host platform. This decreases the need for training and the chances of error by military users within the operational (tactical and strategic) environment, where personnel turnover can be significant. The business environment also has a need for reducing training requirements and human error potential, but this tends to be accomplished more through having the ported HCI applications take on the native platform's look and feel. The user population in the business environment may be accustomed to interacting with the interface style of the particular host platform, and there may be less of a tendency towards personnel turnover. In either type of application, as more HCI applications are developed using UAPI tools, the need continues for designs that support standardization of human performance considerations. The translation of an application should be carefully monitored to avoid the possibility of hybrid GUI styles emerging as an outgrowth of these conversions.

Concepts for UAPIs are currently being developed as industry standards and will represent the basis for commercial GUI development tools that can provide HCI style and application portability. The IEEE, through IEEE P1201.1 and P1201.2, and the NIST have both addressed the issues of GUI application portability by identifying elements of functional commonality for window system objects and by suggesting the use of development tools to provide insulation from and, at the same time, access to HCI style-specific attributes.

Tools that enable transporting GUI applications across host computer platforms represent relatively new technology -- technology that is rapidly evolving and transforming the software development process. It is the range of options and issues surrounding these GUI development tools and their direction of evolution that are of interest to maintaining HCI style consistency and conforming to human performance style guidance as provided in this document.

The developers of GUI applications need to be aware of the current and evolving state of these tools, as the options available from which to choose are varied by commercial product line, each variation holding the potential for even broader utility in future releases.

2.4.2 Range of Approaches to HCI Portability

The need to migrate application software across different user interface styles has driven efforts to separate generic window resource functionality from the style-specific attributes and specific window system access requirements. The notion of separation has prompted a review of the conceptual architecture of both applications and application development environments. This has led to multiple approaches to designing a portable GUI application development tool. The approaches are born out of Object Oriented Programming (OOP) concepts, the way that those concepts lend themselves to the GUI problem, and the characteristics of software development.

GUI systems allow the user to interact with a visual representation of an application model. Visually, the components of an application take on the characteristics of "objects," with all the concepts of OOP lending themselves naturally to the graphical interface application description.

OOP design promotes decoupling interface code from code that implements functionality as well as encapsulation of subproblems and their generic solutions into modules. Together, these two concepts provide the basis for changing the architectural view of the GUI application to one that separates HCI style-defining parameters from general window object functionality, allowing generation of new versions (multiple styles) with the same functionality.

The IEEE committee developing the draft standard for "Uniform Application Program Interface (UAPI) -- Graphical User Interfaces" (P1201.1) -- describes an event-driven "model of interaction between the code implementing application program functionality and the code implementing the user interface," with the fundamental UAPI objects, windows, and events combining to form "pre-built, customizable visual objects" or "controls." This system view establishes layers of functionality to allow for flexible modification, expansion and extension, and orderly communication between layers, all of which are prerequisites for portability. However, the layered view of the HCI application includes more than separating the HCI visual object functionality from its look and feel; it includes the fact that the human-computer interface has some depth, and issues of access and integration exist on both sides of the interface.

The DoD Technical Reference Model (see Figure 1-2) has functions at the application platform level that relate to User Interface Services. These services include the following:

- Graphical client-server operations that define the relationships between client and server processes operating within a network, in particular, graphical user interface display processes. In this case, the program that controls each display unit is a server process, while independent user programs are client processes that request display services from the server. See FIPS 158.
- Display objects specifications that define characteristics of display elements such as color, shape, size, movement, graphics content, user preferences, interactions among display elements. See the *Style Guide*.
- Window management specifications that define how windows are created, moved, stored, retrieved, removed, and related to each other. See FIPS 158.
- Dialogue support services translate the data entered for display to that which is actually displayed on the screen (e.g., cursor movements, keyboard data entry, external data entry devices). See IEEE P1201.x.

Figure 2-4 shows the functions required of an application in addition to its functional code, through its HCI services, and how this relates to the NIST OSE reference model. The OSE model includes the application, the API, the application platform (hardware and software), and the interface to the external environment.

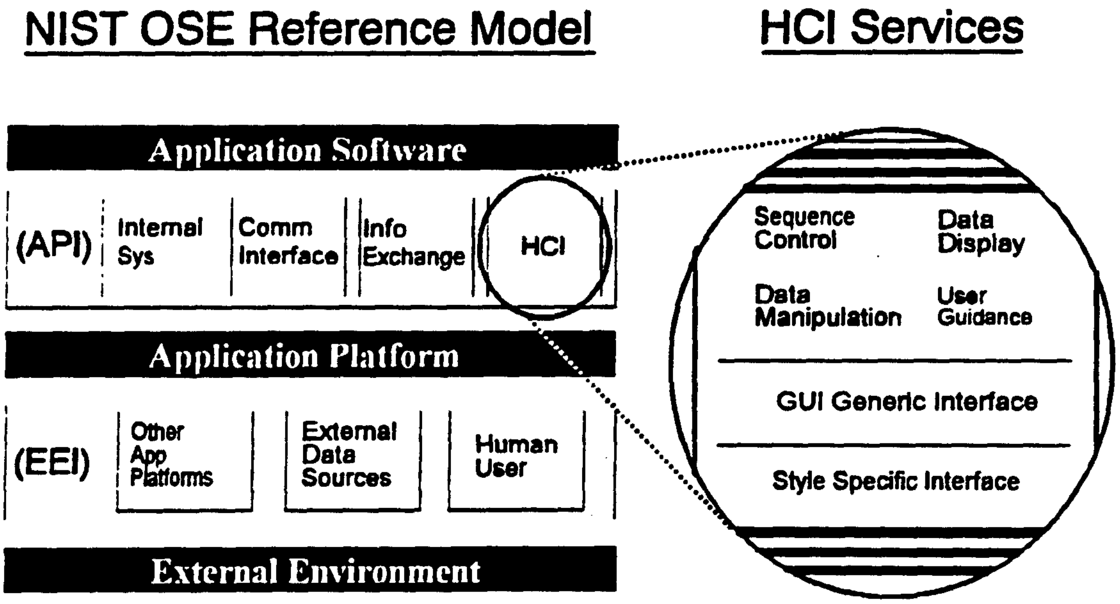


Figure 2-4. NIST OSE Model

The expanded view of the HCI portion of the API depicts an idealized model separating the generic GUI interface object (component) functionality from the specific HCI style's interface and the HCI services of the application code. Identifying and accommodating these HCI services on one side of the interface is equivalent to identifying and accommodating the visual presentation and "direct" manipulation issues on the other side.

The four areas of the human-computer interface services include sequence control, data display, data manipulation, and user guidance, defined in following paragraphs. Since a user interacts directly with these HCI services, they provide part of an outline for the functional requirements of a GUI development tool and road map for evaluating the range of capability such a tool offers.

DEFINITIONS OF THE FOUR AREAS OF HCI SERVICES:

- **Sequence Control** - Sequence control is defined as the actions taken by the user to direct the computer. Actions involve initiating, interrupting, or terminating a computer process and include the system response to the user's action. System responses to an unsuccessful attempt to control the system are included in the user guidance portion of the user human-computer interface. Common methods of sequence control include command line entry, form fill-in, prompted (question and answer) dialogs, menu selection, function keys, and direct manipulation. Alternative methods of sequence control include voice-entered commands and gesturing. Each method has applicability based on user characteristics (e.g., novice, expert, casual, handicapped), function to be controlled, physical environment, available technology, cost, and other design constraints. Most systems employ a combination of sequence control methods in their interface.

- **Data Entry** - Data entry is defined as the act of entering data into the computer and includes the system's response to data entry. The range of user actions covered by this area of HCI is as varied as there are types of data. Text entry is one of the simplest (reference Section 9.0, TEXT). Other forms of data include graphics, maps, imagery (reference Section 10.0, GRAPHICS), and voice (reference Subsection 3.3, Alternate Input/Output [I/O] Devices) -- each with its own method or methods of entry. Other data types include 3-dimensional data (reference Subsection 3.2, Special Displays), multimedia data, virtual reality, and holographic data. Data entry is accomplished using many of the same methods used for sequence control. For example, direct manipulation (reference Section 7.0, DIRECT MANIPULATION) is often used for graphics data entry; form fill-in (reference Subsection 9.2, Form Filling) is often used for textual data entry.

- **Data Display** - Data display includes not only the display of data entered by the user, but the user's ability to control the data display. Thus, text entry into a word processor is a data-entry task, while changing the visual display attributes from normal to bold text is a data display task. Many data display issues are determining factors in the "look" of a system. These include data density, data location, color, contrast, special attributes, image resolution, refresh rate, and update frequency. Similar data display issues exist for audio displays. These include volume, tone, pitch, and timbre. The user's method and flexibility to control the data display portion of an application differs according to the type of data being displayed.

- **User Guidance** - User guidance includes feedback to the user for unsuccessful sequence control attempts (e.g., entering an undefined parameter) as well as guidance for unfamiliar features. On-line help, context-sensitive help, on-line tutorials, and error feedback are all examples of user guidance. Error messages are a portion of user guidance but are usually addressed as part of data sequence control.

The need for a GUI development tool to interact with system services to produce a functional application forms another part of the outline for functional requirements and evaluation. The process required to effect transporting that application from one host platform to another and the anomalies encountered upon porting also need to be considered. This process may use the same or a different GUI style (depending on platforms), and the process is conditional on GUI style.

The IEEE P1201.1 view allows implementation methods to vary in scope and approach. This is done by establishing objects and types of human-computer interaction components. The types of interactions required are those that can form a base set without specifying how those interactions will be implemented or what the components look like to the user.

In addition to the draft standard for a GUI UAPI, IEEE draft "Recommended Practice for Graphical User Interface Drivability" (P1201.2) lists characteristics of GUIs that must be consistent to permit users to easily transfer or switch from one look and feel or application to another without causing confusion, requiring retraining, or provoking errors. The defined uses of mouse buttons; the ability to reduce, enlarge, and close windows through title bar icons; and the changes in appearance of disabled/activated choices are examples of drivability issues.

Commercially available portable GUI development tools have handled access and integration between "layers" in a number of ways. The architectural view of the GUI application may be stratified into conceptual layers of functionality, but tools and kits to develop that GUI application have each targeted different parts of the application development problem. These are exemplified in the following two approaches.

2.4.2.1 Toolkits and Class Libraries

At one level, toolkits and class libraries provide the tailorable GUI components in code form (usually in an interactive graphical form) necessary to build a basic user interface. This speeds development time and helps the developer maintain consistency within the application because it avoids having to construct the individual components from graphics primitives. Extensive programming to link the components into composite/complex window objects and to integrate the HCI interface code into the rest of the application code can be expected with this approach.

And there is no guarantee of portability, unless these components are available in the two-layer form: generic GUI interface object resources and style-specific object attributes.

This toolkit/class library approach can be extended to provide event notifications and query commands to make handling the HCI internal operations (e.g., resizing, moving object location, selection, etc.) conveniently modularized for the developer, saving time and enhancing portability with these generic calls. Modularizing these features and creating generic calls in effect establishes a layered application separating the HCI portion from the application windowing system. To be implemented in the new host window system environment, a library to translate those generic calls to the equivalent host platform calls must be available. Translation libraries must also be available for each type of host platform/HCI style combination supported. This approach is referred to as a "layered API."

Variations on this theme are products that provide a development environment which accesses the native toolkit to create GUI components, and products that supply the equivalent of all native toolkits within the development tool's environment. Some products generate application code templates with the necessary entries to integrate the HCI interface code, and some even assist in integrating data exchange with other software applications/files.

Issues with these products include the following:

- The developer has the flexibility to deviate at will from standard or consistent look and feel within a single application.
- Some toolkit/class library environments do not provide templated code for the API or HCI service links necessary for the application to be functionally complete.

2.4.2.2 Application Framework

A higher level of approach is the application framework: an integrated object-oriented software development system addressing all the application interface services (HCI, Information Exchange, Communications Interface, Internal System) as well as including development tools needed to produce a portable GUI application. Such a system uses the same layered architectural view, but applies it to the entire application development process, not just the HCI portion of the API. This approach is frequently called a virtual API.

This approach allows a developer to work on a level of abstraction that does not presuppose any "common denominator" of native capabilities during design and development of an application, leaving the emulation of attributes not supported by the host system to the API of the application framework. Application design and code are both insulated from the ultimate target application platform with this architecture, so reuse options and portability are a by-product of design.

More than selecting objects from a set of class libraries, the architectural approach to structuring an application with an application framework involves interacting with a flood of structural, window system, operating system, and network system service class managers as well as GUI development service managers. These managers provide the high-level abstraction of services available as well as development tools to build application components. This approach is much more comprehensive and, as a result, covers more of the UAPI issues and provides a greater depth of options for addressing HCI issues.

Very few commercial products use this approach, but that does not represent viability of the concepts, only maturity of the technology in today's commercial products. To date, these products provide a development environment for skilled system programmers. While the GUI resources can easily be designed by anyone with minimal programming background, the application design and GUI resource integration must be specified though calls to the abstraction's version of data types and functions, as well as the notifications and query commands for the GUI -- *not* a matter for occasional programmers.

This approach is much more comprehensive, and as result, covers more of the UAPI issues and provides potential for addressing HCI issues on a broader "open systems" basis to include across networks, platforms, styles, and languages. Other advantages include a much fuller range of functionality and flexibility in GUI layout and development.

Disadvantages include: large amount of overhead code required, very long learning curve, and high cost. Not all HCI services are fully implemented, and there are not as many platforms supported as with the layered products. These tools do not generate code, so the burden of code organization, GUI code set-up and integration, and event processing code implementation falls on the developer. There is also a real danger of hybrid GUI interfaces developing through poor quality control of conversions, the possibility must be reduced by careful compliance reviews.

Intervening levels of approach target a specific combination of system development subprocesses and products to provide a development tool for each approach. Each level of approach brings more flexibility and greater opportunity for portability. But inherent to this flexibility is the risk of inconsistency and the need to ensure that HCI style guidelines are followed.

2.4.3 Environments Supported

Regardless of the level of approach a specific UAPI development tool uses, it has to contend with issues of portability on the HCI Style level and on the development/host platform operating system level. Mappings describing scope of the portability problem and a means to measure the flexibility of commercial tools can be illustrated by example entries in a coverage matrix, such as that in Figure 2-5.

A family of matrixes can be constructed as in Figure 2-5, which describes application portability from one native HCI style to another native HCI style. Because some tools allow hosting an application with an HCI style different than the one native to either the development platform or the host platform's windowing system (or both), there may be several more matrixes to consider.

The five standard graphic HCI styles (Open Look, Motif, MacIntosh, Windows, and Presentation Manager), the native windowing system HCI style, and the operating system/application platform coverage categories provide the dimensions of the coverage matrix.

Dev\Host	UNIX (Motif)	UNIX (Open Look)	UNIX (Native)	PC (Windows)	PC (PM)	PC (Mac)
UNIX (Motif)	X	X		X		
UNIX (Open Look)	X	X		X		
UNIX (Native)						
PC (Windows)	X	X		----		
PC (PM)					----	
PC\ (Mac)						----

Figure 2-5. Example of Native to Native HCI Style Coverage Matrix

For example, an application with a specified HCI style of Motif could be developed on a MacIntosh and ported to a Sun workstation or a PC running Windows 3.1. This coverage feature would show up on a matrix listing the variety of development platform/HCI styles versus the same variety of host platforms, but with Motif listed as the HCI style in each case.

By category, there are tools providing UNIX/HCI style to UNIX/other HCI style portability, UNIX to other operating system (OS) portability, and many OS/HCI style to many OS/HCI style portability. Specific application portability requirements will determine the type of tool best suited for a particular development project. However, the issues to consider are that not all tools have the same coverage and that vendor-stated "coverage" may cause the developer to infer capabilities not intended or not available from a particular product. In addition, certain HCI styles have options not directly transferable to other styles. An investigation to establish suitability of a particular tool might be required to ensure conformance with the native style of the GUI as defined in the relevant commercial style guide and tailored as appropriate in the domain-level style guide. The DoD style guide is not the relevant document against which suitability of a tool should be assessed, because it is native-style-neutral.

2.4.4 Considerations for Use of UAPI Tools

Beyond the issue of coverage, a host of additional considerations, options, and issues should be weighed prior to making a UAPI development tool decision. The use of Ada is mandated by DoD policy. This section frequently refers to "C" and "C++;" however, this does not reflect upon or change DoD policy concerning Ada. Some of these considerations are listed below and discussed in the following paragraphs.

UAPI Tool Selection Considerations:

- Portability Implementation/Features
- GUI to Application Code Interfacing
- Interface Requirements to Existing Software
- Window Object Customization
- Tool/Feature Availability/Maturity
- Costs of UAPI Design Tool(s)
- Runtime License Costs
- Designer/Programmer Training Requirements.

2.4.4.1 Portability Implementation

Ideally, an application developed with a portability tool could be introduced to any covered platform and automatically adapt itself to the native OS and windowing system requirements. Once again, ideally, the option of selecting an alternate HCI style could also be made at this time. The state of technology is not quite to that point yet; most of today's tools require the application to be compiled for a specific target platform before the application can be executed in a new host environment. With the exception of application framework systems, the HCI style is not selectable at runtime.

Most layered (toolkit/class library) approaches require platform-specific libraries to be purchased separately and linked or bound to the application, along with development tool-produced resource files for the user interface. Figure 2-6 shows two such layered approaches for building a portable UAPI application capable of being transported to another platform environment. At the top, the native libraries are emulated by the UAPI tool and at the bottom, the native libraries are used directly.

The process involves developing the window object resource files through either an interactive graphical tool or writing the code by hand. The resource file is then associated with the platform-specific libraries (native to the host platform on the lower approach shown in the diagram in Figure 2-6, and tool-supplied on the upper approach) and the functional code. The process is then completed in each case by making the resource compiler-produced binary files available for the application.

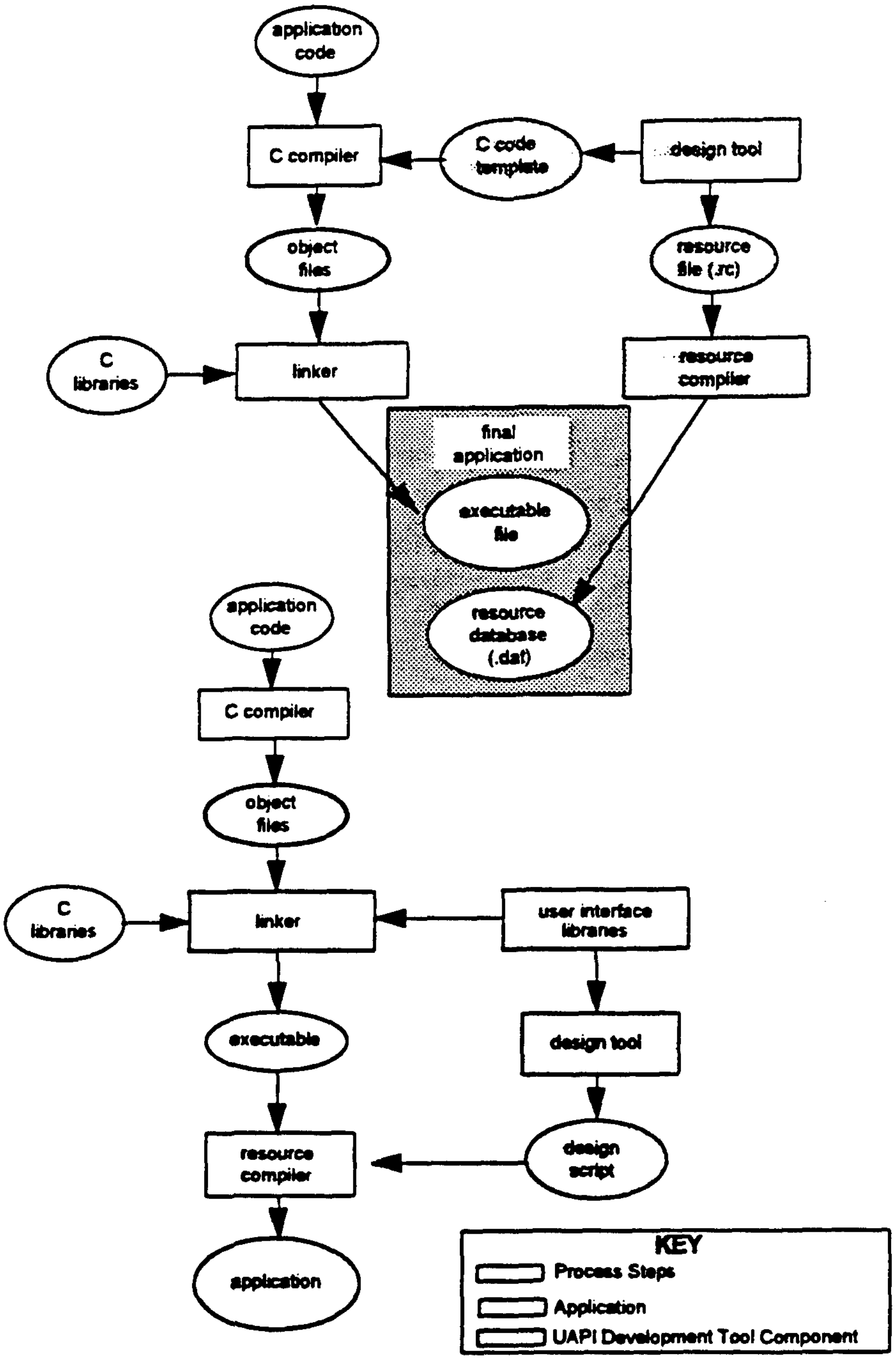


Figure 2-6. Building a Portable UAPI Application

With both methods, application code must be developed beyond what the UAPI development tool generates. In one method, the development tool produces a C-code template outlining setup and prototyping code for the window object resources, event structure, and other associated files to be completed by the developer outside of the UAPI tool. In the other method, all but the callback specifications are generated, and these may be inserted into the code without leaving the

UAPI tool. Both methods separate the generic functionality of the resources from their HCI style-specific attributes until compiling application/ resource files with style specific libraries. The final application has no real separation of the GUI interface layers.

The choice of target platform and HCI style must be made before or during code preparation and cannot be changed at runtime. However, the end result is the same, as long as there is no need to easily switch back and forth.

Application framework approaches consider a larger context for UAPI application development. Therefore, the designer should allow for HCI styles to be selected at runtime (the separation of layers still apparent), but the target platform must be developer-selected prior to compilation and linking.

Porting the application, once designed and built, can involve specific steps and sometimes additional software products. So the issues here are to find out what the steps and the process are, and ask for specific demonstrations. They vary by product, and product literature does not always provide a clear answer.

2.4.4.2 GUI-to-Application Code Interfacing

There are no options for consideration here: no commercial tools exist to specify the dialogues between the application and the user interface. Scripting languages and code generators exist in forms suitable for general purposes but fall short of useful as a means to develop the application code that modifies or manipulates data/system states in response to GUI events.

To make the UAPI application functional, the code must be written to integrate the GUI and its functionality. User interface technology has not progressed to the point where suitable design tools are available to support functional application code development for the portable GUIs. Some products are planning to provide a scripting language that appears to assist in this regard. If this represents a trend for commercial products, it will be a significant enhancement to these development tools.

Most UAPI development tools support applications using the development languages C or C++ (with specific vendor products recommended/required), with Ada rarely supported. The process of integrating the functional code and the UAPI requires a C programmer familiar with event-notification programming. With the exception of GUI layout, these tools cannot be effectively used by nonprogrammers.

2.4.4.3 Interface Requirements to Existing Software

If an application exists, but the requirement to add a GUI to it or port it to other platforms is new, issues become the following:

- Can portable UAPI code be wrapped around existing code to produce a portable product?

- How much effort is required to accomplish the task in either case?

The answers to these questions are subject to semantics and relative to the developer's programming experience and familiarity with the existing code. All of the UAPI development tools reviewed require the functional code to be developed separately as described above, inserting set-up and prototyping code, and tool-unique calls to invoke and manipulate the tool-produced resources at appropriate points. To attach a newly developed GUI, those same UAPI calls would have to be set up, executed, and closed out at the appropriate points in the existing code. Furthermore, most GUI development tools assume a modular software application where each of the routines represent discrete functions of the application. Since this may not necessarily be the case, the existing code may have to be re-engineered in order to insert the GUI code.

The analogy is closer to integration than to wrapping, and the effort involved could be extensive. Given those comments, however, it is possible to take existing code and modify it for portability, assuming that the existing code is in C or C++.

2.4.4.4 Window Customization

Most UAPI development tools have the option of "subclassing" or creating a new version of an existing window component, which can then be tailored as desired. These tools can also modify the look of a specific object through user overrides of color schemes, icon appearances, and pointer images. Either method allows for window customization and provides the flexibility to extend the window resource set that is supported by the development tool. However, this subclassing for customized objects may limit portability of the application if native capabilities are accessed to design the new features. Some UAPI design tools have utilities to design bitmap graphics and incorporate the customized design into the portable resource set, but not all provide this capability.

Customization may allow violations of style guidance established for the standard HCI styles by creating hybrid interface styles. The use of hybrid styles negates the good human factors and human performance practices outlined in this *Style Guide*. Therefore, this flexibility should be moderated through the domain-level style guide.

2.4.4.5 Tool/Feature Availability/Maturity

Some products are in the process of refining their capabilities, some are evolving to offer advanced capabilities, and some are aspiring towards well defined future goals. With each release, commercial products gain maturity and realize more of their potential. For specific application requirements, it is best to compare and verify implementation methods, because innovative approaches in implementing new technology may save development time and effort or force an application redesign because of limitations or accommodations.

The basic set of window components as defined by the IEEE UAPI draft document is available in virtually all commercial toolkits. Most have expanded the list of available objects, and offer customization as discussed above and some means of interactive graphic development. Project and file management services are generally available, though not necessarily handy in their current form; and a means to author and integrate custom help files provides an advantage for some products. Providing the application framework environment to integrate more than the HCI portion of the API will clearly be an advanced step when fully implemented.

Portability to designated platforms can be demonstrated, but product features and services available today are not always clearly separated from target system capabilities and architecture plans in product descriptions. Also, the level of effort in effecting the transportation of an application to a host platform varies by product. At issue is the need to ask very specific questions to determine both the availability of products and the detailed steps required to port applications across host platforms.

It should be noted that there are development tools aimed at a particular application niche -- databases, spreadsheets, knowledge systems -- which contain utilities for creating user interfaces for their particular product. While the GUI interface is not portable in these cases, the total application may be, and it may offer an alternative development option.

Two specific capabilities of interest to operational users within the DoD seem to be conspicuously missing from most commercial UAPI development tools:

- **Cartographic Functions** - Most tactical software applications include requirements for a set of map displays and manipulation operations, such as zoom and jump to coordinates. Special purpose geographic information systems (GIS) provide the map manipulation capability, but do not necessarily interface well with other information systems. Some DoD-sponsored GUI development tools do offer cartographic classes in addition to standard window objects and graphics, but do not offer an interactive graphic development environment.

- **Security Measures** - Operational military users also frequently need to restrict the access of certain subsets of information to authorized users/terminals. OS utilities can provide various levels of security management services across all applications, but are not necessarily portable and do not necessarily offer security services to applications for selective internal use.

Each of these requirements could be built into an application using a commercially available portable UAPI tool and associated application code, but doing so would negate the ease of development offered by using the pre-built resources. Developers should determine if reusable GOTS applications are available that can provide the desired functionality.

2.4.4.6 Costs of UAPI Design Tool(s)/Runtime License Costs

The range of portable UAPI development tool costs is related to the capabilities bundled together, the architectural approach used for the tool, and the platform on which it will be installed.

For example, a class library/toolkit-based development tool that uses the native libraries might run as much as $2000 for a particular workstation/HCI Style platform, while its corresponding workstation/HCI style development product might run $5000. For each platform/HCI style type

(Sun-Motif, Sun-Open Look, HP-Motif, etc.) to which an application is expected to be ported, the specific or separately priced product would need to be purchased. Add to those costs the value of an interactive design tool ($1500/$3000 for PC-based/workstation based) if not already integrated.

This type of pricing method offers project flexibility because users pay only for the specific portability options they require. However, for a development tool that substitutes its own complete set of HCI style libraries, the cost is several thousand dollars more per copy (the cost of additional features), and an application framework product may cost up to $10,000 per copy (the cost of a handling a larger context). Note that some products require the runtime licenses to be purchased separately, although they can be negotiated for purchase in bulk at reduced costs.

2.4.4.7 Designer/Programmer Training Required

None of the UAPI development tools is intended for nonprogrammers. As noted in this section, most tools target C or C++ development environments and require event-notification programming experience for application development.

Some of the tools handle the set-up and clean-up associated with memory management and window object interactions, and others require the developer to provide the code and insert development library calls appropriately. The sophistication of programmer experience required to complete example applications varies by product, but one characteristic shared by all is that there is a long learning curve before developers are capable of applying and taking full advantage of any of these tools.

Screen layout can be easily accomplished by a nonprogrammer with the GUI design tools available. Some tools require programming to reduce the hazards of inconsistency and nonconformance to the native style as defined by the selected commercial style guide and to the design guidelines in this *Style Guide*.

2.4.5 *Style Guide* Implications

The balance between offering the flexibility of many style-specific features on varying platform environments and maintaining consistency in style is a concern for design management. Where development tools restrict some aspects of mixing HCI styles in the same application, they allow others. All of the tools reviewed accommodate custom-developed window objects.

The emergence of UAPI technology will provide a great boon to HCI application developers. While care must be taken to prevent the creation of hybrid GUI interfaces, the benefits for transitions from existing platforms to new platforms or the benefits to training are extensive. UAPIs give the developer a tremendous degree of flexibility in design through:

- Designing an HCI application on one platform and porting it to other types of platforms
- Retaining the pre-existing design or selecting an interface style that is different from the native interface style of the platform to which the porting is targeted

- Assuming the native interface style when porting.

This flexibility also has a burden, ensuring that performance by the user is not compromised through confusing, unique interfaces and increased training requirements. This requires that the flexibility be moderated, to some extent, through standardization of interface styles.

The *Style Guide* and the appropriate domain-specific style guide should be used within DoD to perform this moderation and standardization. These style guides provide the appropriate guidance and framework to guide the developer in tailoring a generic commercial style guides into an application- or system-specific style guide that addresses human rather than software behavior issues, is directed towards DoD design considerations, and presents a more standardized interface style to the user.

REFERENCES

Paragraph	Reference
2.1.1	Marcus (1992) pp. 187-192
2.4.1	NIST(1991a); NIST (1990b); IEEE (1993a), p. 12, p. 91, p. 93; IEEE (1993b); Valdes (1992a); Hagan (1992)
2.4.2	NIST (1991a); IEEE (1993a) p. 12, p. 90; Murphy (1993); Hagan (1992); Valdes (1992a&b); Goldberg and Robson (1985) pp.1-9; Marcus (1992) p.143-214; Microsoft (1991); Smith and Mosier (1986); Hix (1991); IEEE (1993b) p. viii, p. 1, p.19, p.27-29, p. 35-36, p.49
2.4.2.1	Valdes (1992a); Murphy (1993); Hagan (1992); Meyer (1992); Ga Cote (1992); Karon (1992); Chimera (1993); Hix (1992); XVT (1992); Neuron Data (1992a & b); Visix (1992 & 1993)
2.4.2.2	NIST (1991a); Valdes (1992a); Visix (1992); Karon (1992); Murphy (1993); Chimera (1993); Visix (1993a&b)
2.4.3	IEEE (1993a)
2.4.4.1	XVT (1992); Neuron Data (1992a); Visix (1992)
2.4.4.2	Hagan (1992); Meyer (1992); XVT (1992); Neuron Data (1992a); Visix (1992)
2.4.4.4	XVT (1992); Neuron Data (1992a); Visix (1992)
2.4.4.5	XVT (1992); Neuron Data (1992b); Visix (1993)

This page intentionally left blank.

3.0 HARDWARE

Hardware refers to the computer and all supporting devices that impact the HCI. It is difficult to develop standard guidelines for all possible hardware variations within DoD, primarily because of the differences in user requirements and the variety of hardware already fielded. Hardware requirements can vary extensively, depending on the function being performed by the system. Some systems are actually information management systems and business systems that do not require immediate user response to information available through the interface. On the other hand, real-time tactical display and control systems require the user to make immediate decisions and input commands from the information on the interface. Each system has different hardware and interface design requirements based on its primary function. The designer needs to understand the selected hardware and the primary function of the system being developed to provide an effective HCI.

Subsection 3.1 will highlight the procedures used to communicate with system applications using a pointing device or the keyboard.

The purpose of Subsection 3.2 is to present guidelines relevant to the specification, selection, use, or design of displays other than the cathode ray tube (CRT). Subsection 3.2 has been further divided into subsections, each of which describes current technology, cites advantages and limitations, and presents available guidelines. Five types of special display technology are:

- Flat-panel displays
- Large-screen displays
- Stereographic and 3D displays
- Glare reduction techniques
- Touch interface devices (TIDs).

Subsection 3.3 is entitled "Alternate Input/Output (I/O) Devices." This subsection addresses the area of nonstandard access to a GUI environment. The guidelines in this subsection consider the requirements of the CAP.

3.1 INPUT DEVICES AND PROCEDURES

Subsection 3.1 will highlight the procedures used to communicate with system applications using a pointing device or the keyboard. A comparison is made between Motif and Open Look to illustrate the impact of GUI style selection on hardware. For a more detailed explanation of the input procedures, consult the appropriate GUI style guide.

3.1.1 Pointing Devices

A pointing device (e.g., mouse, trackball, tablet, or light pen) allows a user to navigate rapidly around the screen and to specify and select objects for manipulation and action. Throughout this *Style Guide*, the mouse is used as the reference example for all pointing devices.

3.1.1.1 Mouse Button Definitions

Within the DoD community, both two-button and three-button mice are used. For users who must interact with both Motif and Open Look applications, it is important to note that the button definitions and button use differ. The mouse button operations supported by both GUIs are consistent and can be defined as follows:

- **Press** - pushing the mouse button and holding it
- **Release** - letting up on the mouse button
- **Click** - quickly pushing and releasing a mouse button before moving the pointer
- **Double-Click** - pushing and releasing the mouse button twice in quick succession
- **Move** - sliding the pointer without pushing any mouse buttons
- **Drag** - pushing the mouse button and holding it while moving the pointer.

To "drag an object with the mouse" is to move the pointer over the object, press the SELECT button on the mouse, move the mouse until the object is in the desired location, then release the SELECT button.

3.1.1.2 The Pointer

A key workspace element is the pointer. Objects on-screen can be manipulated by positioning the pointer over an object and appropriately pressing the mouse buttons. The user moves the pointer by moving the mouse on the mouse pad.

> *NOTE: Different actions are used to move the pointer with other pointing devices, such as trackballs and light pens.*

Pointer shapes provide visual clues to the activity within a window. For example, an hourglass or watch-shaped pointer may indicate that an application is busy, and a cross-hair can be used when sighting on a graphics display.

With the exception of applications using computer-controlled tracking, the pointer should remain where it is placed until moved by the user or the application. The pointer should not "drift."

3.1.2 The Keyboard

The keyboard in an operational situation should be virtually interchangeable with the mouse to allow a user to interact with the application by using a pointing device, the keyboard, or both. Business area applications should allow the keyboard to substitute for the mouse but may not find it advisable to provide some keyboard functions through the mouse. Although keyboards vary greatly in number and arrangement of keys, most keyboards include the following:

- **Alphanumeric Keys** - Letters of the alphabet, numbers, punctuation symbols, and text-formatting functions (e.g., Tab, Return, Space Bar)
- **Modifier Keys** - Keys (typically Shift, Control, and Alt) that modify or qualify the effect of other keys (or pointing device inputs) for as long as they are held down
- **Navigation Keys** - Keys that are used to move the cursor (e.g., arrow keys, Home, End, Page Up)
- **Function Keys** - Keys (typically F1 through F10) provided for extra or general functions
- **Special-Purpose Keys** - Keys that have a special function (e.g., Help, Delete, Escape, Backspace, Insert, and Enter).

Because keyboards differ and function keys vary according to application and GUI, a function should not be solely available through a function key. Guidelines for commercial style guide application key assignments are provided in the respective GUI style guides.

3.1.3 Window Input Focus

Usually, several application windows are ready to accept input; but only one window, the one with "input focus," actually receives the user input. The window with input focus is known as the active window and is the window where keyboard input appears and pointing device inputs apply.

Most interfaces provide explicit input focus; that is, the user (or application) performs an action (e.g., types appropriate keyboard accelerators, clicks a pointer inside a window, or moves a window to foreground through menu selection) to assign input focus. Implicit focus (the focus is automatically assigned to the window containing the location cursor) is often provided as an option. The default for applications should be explicit focus.

A window with input focus should move to the front of the workspace and be highlighted in some fashion, such as highlighting the window frame or title bar.

3.2 SPECIAL DISPLAYS

The CRT is the principal display technology used in computer-based systems. Success of the CRT can be attributed to its ability to inexpensively deliver full-color imagery at high luminance and resolution. However, tasks of the modern military and emerging alternative display technologies have permitted development of computer-based systems using display technology other than the traditional CRT. Examples include the liquid crystal display (LCD) used in portable computers, and projection technology used to brief military personnel in command centers. As a result, the designer has more alternatives when selecting a display for a military system. The purpose of this discussion is to present guidelines relevant to the specification, selection, use, or design of displays other than the CRT.

Subsection 3.2 is divided further into subsections, each of which describes current technology, cites advantages and limitations, and presents available guidelines. Many reports upon which this subsection is based were published in various conference proceedings rather than in refereed scientific journals and, therefore, may have had less extensive peer review and professional scrutiny. The *Style Guide* user should consult the references for further information on display technologies and human performance considerations.

3.2.1 Flat-Panel Technology

A flat-panel display is flat and light and does not require a lot of power. "Flat" means being thin in form, as well as having a flat display surface. An ideal flat-panel display has the following characteristics: thin form, low volume (cubic size), even surface, high resolution, high contrast, sunlight readable, color, low power, and light weight. Recent advances in flat-panel display technologies have made them realistic alternatives to CRTs for displaying information at computer workstations. Advances have been made in many areas: addressability, contrast, luminance, and color production. Continued research in flat-panel displays has resulted in introducing high information content products that challenge the CRT in specialized applications.

Although there are many different types of flat-panel display technologies, LCDs, electroluminescent displays, and gas plasma displays are the only flat-panel technologies currently mature enough and economical enough to be used in DoD. A major characteristic of each of these display technologies, as distinguished from CRT technology, is that images are formed by turning discrete, non-overlapping, rectangular, cell-based pixels on and off. This discrete, pixel-based structure provides part of the reason that measures of image quality used to evaluate CRT resolution cannot be effectively used to predict image quality and human performance with flat-panel displays.

Factors affecting human performance that differ from the guidance given for CRTs include character-to-character spacing, interline spacing, character and symbol design, the effect of ambient illumination, image polarity, and failure mode. An overriding guideline when specifying flat-panel display technology relative to the CRT is to apply more stringent image quality criteria when selecting flat-panel technology.

3.2.1.1 Character Size

Character size is an important variable affecting performance error rates. Height and width of the character and the size of the pixel matrix have important effects on human performance. Exercise special care when determining the character size to use on a flat-panel display.

a. To improve text search and sorting task performance, use a 9 x 13 pixel matrix or larger.

b. When displaying dot matrix symbols in nonvertical orientations, use at least an 8 x 11 pixel matrix and preferably a 15 x 21 matrix size.

c. Character stroke width (SW) should be in the range defined by: (character height ÷ 12) + 0.5 SW (character height ÷ 6). See the following list for guidance.

Pixels in Upper Case Character Height	Minimum Stroke Pixel Count	Maximum Stroke Pixel Count
7 to 8	1	1
9 to 12	1	2
13 to 14	2	2
15 to 20	2	3
21 to 23	2	4

d. Character height to width should be in the range defined by: (character height x 0.5) character width (character height x 0.9). See the following list for guidance.

Pixels in Upper Case Character Height	Minimum Width Pixel Count	Suggested Minimum Width Pixel Count	Maximum Width Pixel Count
7	4	5	5
8	4	6	7
9	5	6	8
10	5	7	9
11	6	8	10
12	6	9	11
13	6	9	12
14	7	10	13
15 or 16	8	11	14

3.2.1.2 Luminance Nonuniformity

Display luminance should be uniform across the surface of the display. Maximum luminance nonuniformity levels should be consistent with the values specified as follows:

Test Object Separation	∠higher Maximum
At the Design Viewing Distance	∠ lower
>7°	1,7
5 to 7°	1,6
4 to 5°	1,5
2 to 4°	1,4
2°	1,3

3.2.1.3 Image Formation Time

Image formation time (IFT) is the time required to render a new image. Four classes of IFTs (see below) have been defined, each relating to information-update requirements for the application. IFTs for all systems should be consistent with Classes III and IV.

Class	Image Formation Time in Milliseconds	Significance
I	120<t	Satisfactory for displays that update an entire page of information at once. Noticeable during key entry. Applications using scrolling, animation, and pointing devices are significantly degraded.
II	55<t 120	Satisfactory for displays that update an entire page of information at once. Not noticeable during key entry. Applications using scrolling, animation, and pointing devices are somewhat degraded.
III	10<t 55	Satisfactory for most applications. Motion artifacts can be distracting but are usually acceptable.
IV	3<t 10	Motion artifacts become less noticeable at formation times approaching 3 milliseconds.

3.2.1.4 Display Failures

The three most common failures on matrix-addressable displays are cell failures involving individual elements, vertical line failures, and horizontal line failures. Displays can fail actively or passively and leave pixels or lines permanently on or off, respectively.

- Because cell failures often lead to greater performance problems, select displays that minimize the likelihood of cell failure.
- To minimize the performance impact of cell failures, select displays and set display polarity so these failures are likely to match the display background.
- When display element failure is an expected problem, increase the redundancy in the text to minimize the impact on reading performance associated with display element failures.
- Recognition and identification performance with cartographic display is subject to significant decline with as little as 1% pixel failure. Select and maintain displays to ensure a pixel failure incidence below this level.
- Use characters with a pixel matrix larger than 7 x 9 pixels in order to reduce the negative effect of "on" failures.

3.2.1.5 Polarity (Contrast)

A display with white (or light) characters on a black (or dark) background is said to have "negative contrast" or to be a "positive (image) display." Conversely, dark characters on a light background are said to have "positive contrast" or to be a "negative (image) display." If character stroke width, modulation, and luminance values are nearly equal for both polarities, select a positive contrast/negative image display for better reading speed, search time, and search error-rate performance. The presentation of dark characters on a light background may reduce the effects of reflections on the surface of the display. The effects of glare caused by superimposed reflections are the same for displays of either polarity.

3.2.2 Liquid Crystal Displays

LCDs are perhaps the most developed and popular flat-panel display technology. Rather than emit light, as do active flat-panel technologies, an LCD controls or modifies the passage of externally generated light. An LCD is typically made of transparent plate electrodes that sandwich a liquid crystal substance. Voltages applied to these electrodes cause realignment of the liquid crystal material, changing its optical properties and allowing light to propagate through the material. By selectively applying voltage to the electrodes, individual display elements can be made light or dark to create the desired image on the LCD.

The LCD is available in a large variety of formats for both commercial and military applications. Display size and resolution range from small, character-based displays (e.g., those in watches) to full-screen computer displays with resolutions to 640 x 480 pixels. LCDs can be monochrome or color and may operate with backlighting across a wide range of ambient illuminances. LCDs are especially suited for information display in environments where ambient illuminances are high.

Advantages of the LCD include excellent contrast, long life, rugged design, low voltage, and low power consumption (except when backlit). LCD technology is limited by slow speed, limited color capability, temperature range, and manufacturing problems for larger panels with higher resolution.

3.2.2.1 Ambient Illumination

Provide adequate levels of ambient illumination, because reading performance improves as ambient illumination increases over the range 20-1500 lux.

- Consider LCDs for effective display in high ambient illumination situations.
- In low light situations, provide the ability to adjust the viewing angle and the amount of backlight to enhance the legibility of presented information.

3.2.2.2 Polarity (Contrast)

For legibility of transmissive or backlit LCDs, use dark characters on a light background (positive contrast/negative image displays). For reflective LCDs, use light characters on a dark background for better performance.

3.2.2.3 Level of Backlighting

Minimize or eliminate use of backlighting because display reading errors increase as the level of LCD backlighting increases over the range 0-122 candela per square meter (cd/m^2).

> NOTE: *cd/m^2 or nit (normalized intensity) is a metric unit for reflected light. One cd or nit is equal to 0.29 footLambert (fL) or one fL is equal to 3.4 nit. A fL is a unit used to measure light reflected from a surface. An ideal surface that reflects all light striking it and diffuses it with perfect uniformity has a luminance of one fL when illuminated by a one footcandle source.*

3.2.2.4 Backlighting and Angle of View

Carefully consider the potential impact of user performance decrements before using a backlit LCD that is to be viewed off-axis. Backlighting impacts user performance adversely when the display is viewed at an angle.

3.2.3 Gas Plasma Displays

Plasma panels or gas discharge displays are a widely used flat-panel technology in the information and computer industry because of their inherently high-contrast and high-resolution capabilities. Images are formed by ionizing a gas, usually neon, trapped between a set of horizontal and vertical electrodes. When an electrical field created by the electrodes is increased rapidly, the gas begins to discharge, resulting in a glow that forms an image. The image can be maintained by sustaining the electrical field, or erased by dropping the voltage below some threshold value. The high contrast exhibited by plasma panels is a result of almost no light output in the "off" state (the electrical field is below threshold) and high luminance in the "on" state. Full-color plasma can be made by depositing phosphors on the glass display surface. The plasma gas discharge in this case excites the phosphor, in much the same manner the electron beam does in a CRT; and color images are produced.

Plasma panels can be found as either monochrome or full-color displays in a number of sizes and configurations. A major advantage of plasma technology is that very bright, high-resolution panels are available. Panels that measure 2048 x 2048 pixels at 100 pixels/inch are available, as well as those that can be viewed in direct sunlight. Panels with luminances of 150-600 cd/m^2 have been produced, with typical large-area, high-resolution display luminances being 30-50 cd/m^2. Full-color direct current (DC) plasma panels are not yet able to achieve the luminance output nor the display life normally associated with plasma technology.

Features of plasma technology generally include uniformity, high resolution, large size, long life, ruggedness, and absence of flicker. Applying plasma technology in the computer and information industry is limited by high voltage and power requirements, complexity of the drive circuitry, low luminous efficiency, need to develop more fully a color capability, and lack of developers of the technology. Limited information on interface performance is currently available in the literature. In the absence of specific guidance, the designer should use the most conservative approach to interface design.

3.2.3.1 User Concurrence

Verify the use of gas plasma displays by user personnel as a viable alternative.

3.2.3.2 Testing Prototypes

Designers and developers of computer-based systems should consider field testing prototypes before committing to gas plasma technology.

3.2.4 Electroluminescence (EL) Displays

Electroluminescence (EL) displays consist of a layer of polycrystalline phosphor powder or evaporated film phosphors sandwiched between sets of vertical and nearly transparent horizontal electrodes. When an electric field is applied across the polycrystalline phosphor, it is stimulated and light is emitted. The display resolution and the shape of the pixel are defined by the arrangement of the electrodes. EL displays are usually classified as one of four types: either alternating current (AC) or DC thin-film displays, or AC or DC power displays.

AC thin-film displays and DC powder displays are the most advanced of the EL technologies, and discussions of the strengths and weaknesses of EL technologies will be limited to these types. AC thin-film EL displays have very good luminous efficiency, high contrast, good resolution, and long life. As with some of the plasma technology, these displays require high voltages and complex drive electronics, are expensive, and need research and development to deliver full-color performance. Currently, ELs that display up to 864 lines by 1024 pixels are available for alphanumeric and graphics applications. DC power ELs have a good appearance, simple structure, good luminous efficiency, and the ability to produce gray scales. However, they require high voltages and complex drive circuitry, and have limited luminance output coupled with high reflectance, which may lead to contrast problems. Also, they are expensive. Full-color displays have been produced using DC powder technology, but this area too needs more development. Currently, resolutions compatible with graphics interfaces for personal computers are available (480 lines by 640 pixels) for use in applications that require alphanumerics and moderate graphics. The greatest problems with EL display technology are that developers are few, and investment in research and development is small.

Interface performance information on EL displays is not available in the current literature. In using EL displays, follow the guidelines recommended for LCDs, and use the most conservative level of those recommendations.

3.2.4.1 User Concurrence

The use of EL displays should be verified by the user personnel as a viable alternative.

3.2.4.2 Demonstration of Concept

The use of EL displays should be field-prototyped before incorporating into a new system.

3.2.5 Glare-Reducing Techniques

Glare, as observed on the face of an electronic display, is composed of two components. 1) Diffuse glare or veiling glare, caused by the general illuminance in the environment, can be characterized as a field effect and has little or no modulation. The effect of diffuse glare is to reduce the effective contrast of the display. 2) Modulated or specular glare is the first surface reflection off the faceplate of the display and results from some object or objects in the area surrounding the display. The effect of this type of glare is the appearance of unwanted images on the display surface, making the displayed information more difficult to see and interpret.

The most effective control of glare is to design appropriate workspace illumination so neither diffuse nor specular glare is produced. This is the only method of glare control that will not compromise the resolution and contrast of the display. Because it is not always possible to properly control the sources of illumination, glare reduction techniques have been developed to minimize the unwanted effects of glare.

Many kinds of glare control techniques are used in the electronic display market. Some are screen meshes placed over the display surface, chemical or mechanical etches of the faceplate of the display, anti-reflective coatings, and bonded quarterwave filters. Each has advantages and disadvantages in terms of ability to control diffuse or specular glare, and in terms of effect on display resolution, flexibility, maintenance, and cost. For example, a bonded quarterwave filter only minimally degrades display resolution but is very expensive, whereas a mesh overlay is very inexpensive but has a major effect on display resolution.

The effectiveness of the glare reduction technique is a function of its ability to suppress each component of glare, while minimizing degradation to the display's resolution and contrast. The desired effect is to match as closely as possible the display performance under optimum conditions. While both contrast and resolution are degraded in an absolute sense, the effective image quality in the operational environment and the acceptance of the display system should improve.

Selecting the glare control alternative most effective for a particular display depends on the information to be displayed, task required of the operator, and environment in which the display will be used. In a command and control facility, use careful analysis and testing to determine the type of glare reduction measures that should be taken.

3.2.5.1 Reflected Glare

When possible, avoid reflected glare by altering the angular relationship among the observer, display, and glare source. For example, provide the ability to adjust height, viewing angle, and/or contrast.

3.2.5.2 Filter Selection

When possible, leave selection of the specific glare-reduction technique to individual users.

3.2.5.3 First-Surface Specular Reflections

Because many types of flat-panel display consist of multiple plates of glass, each of these acts as a specular reflector. All flat-panel displays should incorporate a first-surface treatment to diminish first-surface specular reflections.

3.2.5.4 Etched Filters

Etches with gloss values of 45 or less should not be used on monochrome CRTs, and etched filters should not be used at all with high-resolution displays.

3.2.5.5 Projection Displays

With projection displays, minimize glare potential by positioning projection equipment so the light source is not readily visible to viewers.

3.2.6 Large-Screen Displays

The DoD operational environment not only imposes requirements for information display at individual workstations, but also for work areas where many persons must observe and use the information presented on a display. Large-screen displays first appeared in the operational environment as the presentation of mission information on transparent Plexiglas overlays. Today, large-screen technology found in the command center is computer-driven, with the ability to present graphic and video information.

Large-screen display of surveillance, weather, and intelligence information to operations personnel and as a briefing aid to the principal decision-makers is typical in the operational environment. Unfortunately, because of minimal brightness, poorer contrast, and lower resolution when compared to higher resolution desktop displays, most implementations of large-screen technology have been disappointing to the users. Current display technology offers the military a number of choices when implementing a large-screen display. Figure 3-1 illustrates the types of technology available for fielding large-screen displays. The military currently uses both direct view (e.g., high luminance CRTs or large plasma displays) or projection (e.g., light valves or projection CRTs) large-screen displays in its operational facilities.

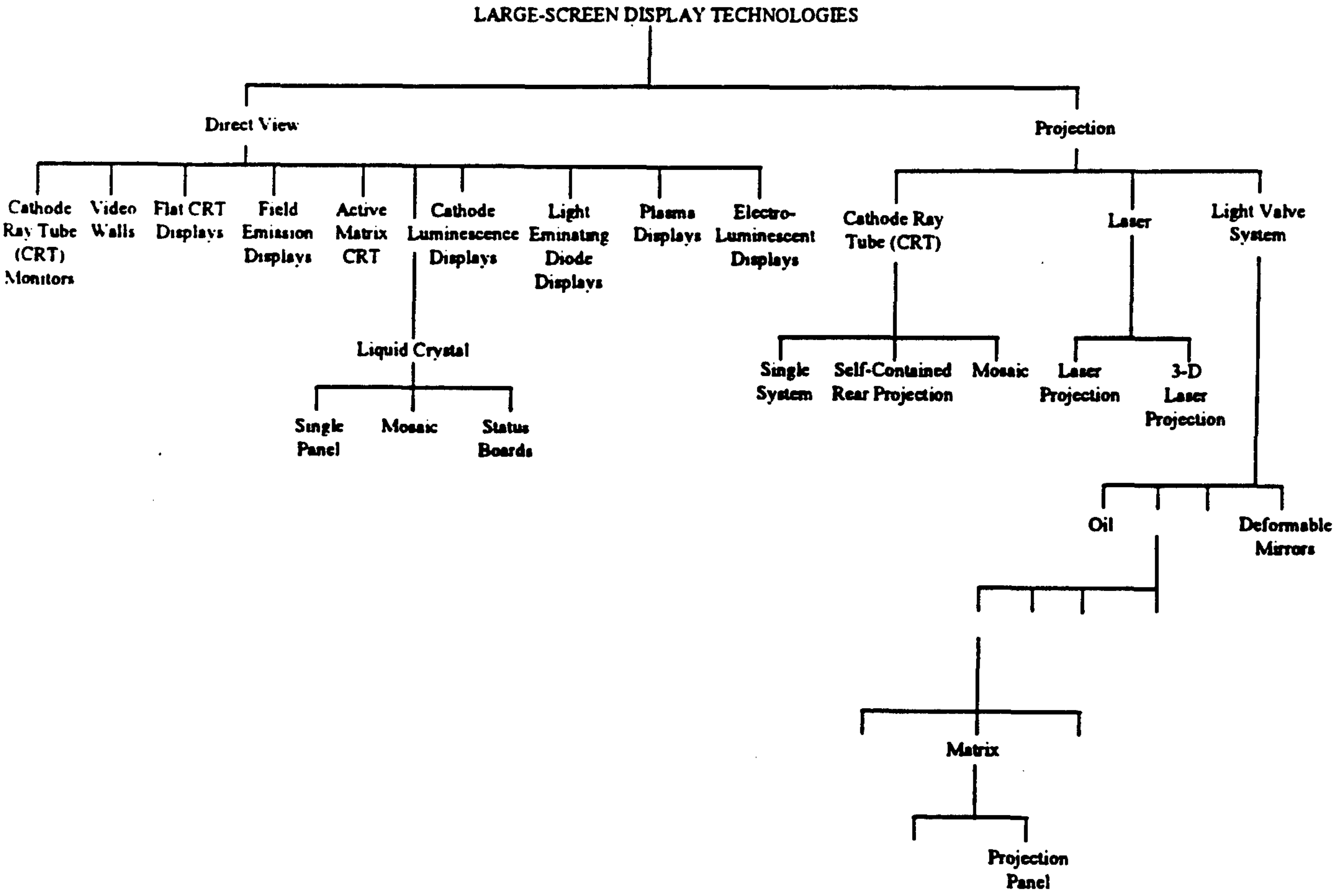


Figure 3-1. Large-Screen Display Technologies

Requirements for large-screen display selection in the operational facility vary relative to use and size of the room in which the display will be used. Large-screen displays found in small briefing rooms (approximately 600 square feet) have a screen size of about 50 square feet, horizontal resolution of 300-1000 pixels, and a luminance output of 300-500 lumens. The briefing room must be nearly dark (<2 foot candles [fc]) when the large-screen display is in use.

The other type of facility where large-screen displays may be found is in the command and control center. The command center may be as large as a two-story structure larger than 2400 square feet, where the information is presented to 10-100 personnel. Large-screen displays in this environment have a screen size of about 100 square feet, screen luminance of 10 fL, horizontal resolution of 800-1000 pixels, and a luminance output in the range of 1000 lumens. Room illuminance in the command center is adjustable from about 5-15 fc, but a more normal office illuminance of 75 fc is often requested. Full-color capability is required of large-screen displays in both facilities. Large-screen displays that are larger, brighter, and have more resolution are desired for the command and control environment.

Selecting or designing a large-screen display, especially a projection display, may be more complex than for other workstations. The effects of ambient illuminance, observer location, and type of data to be displayed are critical in implementing large-screen display technology. Presentation requirements for data not only relate to one's visual acuity for symbol size and contrast when dealing with projection technology, but also to screen size, screen format, symbol luminance, and screen gain. For example, as screen gain increases, the ability to view the screen off the center line decreases. However, a certain amount of screen gain may be necessary to present an image of the necessary contrast, given the expected ambient illuminance in the room. In addition, symbols, graphics, and text should be designed to compensate for the degraded viewing conditions that may exist in the operational environment due to a number of factors. Implementing large-screen displays in the operational environment should always take into account the environmental factors, as well as the information display requirements.

> *NOTE: Typical office ambient level is greater than 75 fc, whereas typical command and control centers are 5-15 fc.*

3.2.6.1 Character Dimensions

Because information is often viewed off the center axis, use character sizes between 10 and 20 minutes of visual arc, with a minimum of a 10 x 14 dot matrix format. When legibility is important, the minimum character height should be 16 minutes of visual arc.

3.2.6.2 Stroke Width

Ensure that the ratio of character stroke width to character height is 1:6 to 1:10. Use characters with double stroke widths in situations requiring off-axis, longer distance, and/or viewing under difficult lighting conditions.

3.2.6.3 Luminance

Ensure that modulated output luminance, spatially averaged over the full screen, is 300-400 lumens for small conference rooms and command posts and 750-2000 lumens for a command center, assuming 20-40 fc ambient lighting in each case.

3.2.6.4 Size versus Luminance

To ensure legibility, small characters (5 min. arc) require contrast ratios of 15-20:1, and large characters (>20 min. arc) require contrast ratios of 1.5-5:1.

3.2.6.5 Aspect Ratio

Aspect is the ratio of horizontal to vertical dimensions of a character or image. Ensure that character aspect ratio is approximately 1.33:1.48 (width:height ratio).

3.2.6.6 Modulation Depth

Ensure that a display delivers at least 15% visual contrast when measured as modulation depth $[(L_{max}-L_{min})/L_{max}]$, when an alternating pixel pattern is displayed at normal luminance levels.

- Ensure that contrast ratio between the reflected luminance of the screen with a projected light source and the reflected luminance of the screen without a projected light source is approximately 500:1.

- Use positive contrast (black characters on a white background).

3.2.6.7 Displayed Data Characteristics

- Avoid displaying too much data. As with standard displays, consider data type, amount, and appropriate sequence of presentation in designing large-screen display screens.

- If displaying color-coded targets, use only a neutral color such as gray for the background.

3.2.6.8 Projection Equipment

- Minimize glare potential by positioning the projection equipment so that it is not readily visible to viewers.

- To minimize optical distortions, ensure that image source equipment and the projection screen are fully parallel. Electronic or optical distortion-compensating devices may be used to compensate for any remaining distortion and to assure clarity of displayed information.

- Consider using rear projection or other direct view large-screen displays when increased contrast demands are encountered and/or when there is a need to position personnel in the field of projection.

3.2.7 Stereoscopic/3D Displays

Displaying 3-dimensional (3D) images and graphics is an emerging technology that may benefit future military applications. Examples where 3D technology may be used are in battlefield and theater of operations analysis, photo-interpretation, teleoperation, air space control, and training and simulation exercises. The goal of proposing 3D displays is to improve user performance and increase naturalness of the interaction. Most current 3D technology is experimental and, as such, is not suitable for an operational environment, although a few stereographic and true 3D electronic displays can be purchased commercially.

Because traditional display technology is a 2-dimensional medium, it has not been able to take full advantage of the human visual system to interpret complex spatial data. Binocular depth information, such as vergence or horizontal disparity, normally in the scene, are not available in the traditional electronic display. Compensation for this has been accomplished by using monocular cues, such as interposition, shading, and perspective. However, improvements in naturalness of the display and potential for gains in human performance with computing systems have stimulated development of systems that make use of the stereoscopic capabilities of the human visual system.

Three-dimensional display technology is classified as stereoscopic and autostereoscopic. The major criterion distinguishing stereoscopic displays from autostereoscopic displays is that the latter requires no special viewing aids to see the 3D image. There is also a difference in the amount of information necessary to create the 3D image.

Stereoscopic displays create a 3D image by requiring an observer to wear a pair of glasses that provides separate images to the left and right eyes. When alternate fields are presented to the eye sequentially at the appropriate rate, the illusion of depth is created. The temporal phase difference that accounts for the stereopsis creating the 3D illusion is usually implemented by requiring the viewer to wear a pair of glasses containing either shuttered lenses, polarized lenses, or red and green lenses. By synchronizing the image presentation to the operation of the glasses, images corresponding to the left and right scenes are presented to the viewer and the illusion of depth is created.

Autostereoscopic displays, by contrast, can be viewed directly. These displays generally use a multiplanar approach to add depth to a 2-dimensional image. Examples of this type of 3D display are BBN's SpaceGraph 3D Display System (uses a flexible mirror to provide the z axis), Tectronix liquid crystal shutter 3D display (uses LCD technology together with a CRT display to create a 3D effect), and Texas Instruments' Omniview (uses a rotating multi-planar surface to produce the z axis). Holography has also produced 3D images, but none to date has been created in real-time.

While innovative technology to provide 3D images is becoming available, no clear guidance outlines where stereoscopic displays might best affect task performance or subjective image quality. Additionally, no database derived from applied vision or human factors research currently exists for developing application guidelines for this new technology. Consequently, the system designer must be cautious in applying 3D display technology in the military environment. Current technology often limits field of view, number of observers, and type of data that can be presented. It also may exacerbate visual deficiencies that normally have little effect on task performance. Guidance presented here is by no means complete. Many questions remain unanswered, both in terms of human visual response to artificially generated depth from electronic displays and the ways best to enhance performance using this technology.

3.2.7.1 Purposeful

Presenting 3D information must be purposeful to benefit the user. That is, 3D displays should be associated with the type of work to be performed and required for task completion.

3.2.7.2 System Performance

Presenting 3D or depth information should not slow information updates, degrade other aspects of system performance, or degrade image quality.

3.2.7.3 Interocular Crosstalk

Interocular crosstalk or bleed-through occurs in stereoscopic displays when images intended for the left eye are seen by the right eye and vice versa. Because this compromises the observer's ability to fuse the image and perceive it as a 3D object, ensure zero interocular crosstalk between the two images.

3.2.7.4 Color Coding

Avoid saturated primary colors, as these colors may evoke depth perceptions that may be inconsistent with stereopsis, affecting the perception of depth. Designers should use secondary colors rather than saturated primary colors in coding stereoscopic images.

3.2.7.5 Symbols

When displaying symbols, ensure that disparity ranges from 0 to 20 minutes of arc in both crossed and uncrossed directions.

3.2.7.6 Dynamic Depth Displays

When using dynamic depth displays, ensure that the temporal modulation of stereopsis is approximately 1 Hertz (Hz) to ensure the most accurate perception of stereo-motion.

3.2.7.7 Depth-Coded Objects

Spatially separate depth-coded objects in stereoscopic images to eliminate disparity averaging, crowding, or repulsion.

3.2.7.8 Size Scaling

Scale image size to improve the perceived disparity of the image. When accurate size perception is critical to task performance, scale image size for an individual observer.

3.2.7.9 Luminance

Because brightness is also a depth cue (bright objects are viewed as nearer), co-modulate luminance with stereopsis.

> *NOTE: Display parallax is the apparent displacement of an object displayed on a curved CRT screen and viewed through a flat touch interactive device (TID).*

3.2.8 Touch Interactive Devices

A TID is an input device that permits a user to interact with a system by pointing to objects on the display. The TID is considered here because some implementations of touch technology can severely degrade quality of the displayed image. Degradation in image quality using TIDs may be a result of decreased display luminance, reduced display resolution due to visibility of conductors or the device material, increased susceptibility to glare, and dirt on display surface as a result of touching the display surface. Display parallax, caused by separation between touch surface and touch targets, may also contribute to problems with implementing TIDs.

There are six basic types of touch-screen display technologies, each having an impact on display parallax, transmissivity of light, and glare. Each is briefly discussed below. The designer needs to be aware of advantages and disadvantages of each type of TID when selecting hardware and designing interfaces using TIDs.

- Fixed-wire TIDs place wires, either in parallel or in grid fashion, in front of the display. Finger contact with the wire(s) signifies the x,y coordinate of the user's response. This technology is associated with minimal parallax, 70-80% transmissivity, and a medium to high degree of TID glare.

- Capacitive TIDs consist of a transparent conductive film on a glass overlay. Touching this surface changes the small electrical signal passing through the surface, and this signal is converted into the corresponding x,y coordinate. This technology is associated with minimal parallax, 85% transmissivity, and a medium degree of TID glare.

- Resistive membrane TIDs are "sandwich" devices in which a touch results in the contact of two conductive layers. Specific current and voltage levels are associated with individual x,y coordinates. This technology is identified with minimal parallax, 50-60% transmissivity, and a high degree of TID glare.

- Infrared (IR) or light-emitting diode TIDs use IR transmitters along two perpendicular sides of the display frame and photocell receptors along the opposite sides of the frame. A user touch breaks the resulting matrix of light beams, and the appropriate x,y coordinates of the touch are thus determined. This technology is associated with no parallax problems in seeing the display (although a noticeable degree of parallax exists between the plane of the IR grid and the screen surface for touch responses), 100% transmissivity, and no TID-related glare.

- Surface acoustic wave TIDs operate in similar fashion to IR TIDs, except that the matrix overlay is one of ultrasonic sound beams rather than IR beams. Another approach, "reflective array," uses a piezoelectric transmitter and a series of reflectors and receivers. Touch x,y coordinates are determined by differential timings in reception of the acoustic waves. At least some devices require glass overlay screens. This technology is associated with minimal parallax, 92% transmissivity, and a medium degree of TID glare.

- Pressure-sensitive devices use strain gauges mounted between the display screen and an overlay. Output voltages of these strain gauges are encoded into the appropriate x,y coordinates. This technology is associated with minimal parallax and zero TID glare. Figures for transmissivity are not applicable because the overlay is built into the display screen.

3.2.8.1 Parallax

Minimize TID/display parallax because it has been shown to lead consistently to poorer entry time and touch count performance.

3.2.8.2 Specular Glare

Minimize specular glare for applications using TIDs.

3.3 ALTERNATE INPUT/OUTPUT (I/O) DEVICES

The focus of HCI design literature and research has been on the software, displays, physical environment, and computer equipment aspects of the interface. Approaches to testing and evaluating HCIs are usually based on the machine rather than on the human portion of the computer interface. Perceptual characteristics of the expected user are rarely investigated, and interface design ignores known population perceptual limitations. Using color to transfer information does not take into account the potential of color-deficient vision problems in user populations. Using auditory codes does not take into account potential hearing deficits by frequency and adjust outputs according to known population characteristics. The distribution of visual acuity within the user population is usually not considered. It is more likely that environmental impacts on the system will be defined than will user perceptual characteristics.

Accessibility of computer-based systems by persons with disabilities is U.S. Government policy based upon Public Law 99-506 and Public Law 100-542. Individuals with limited hearing, vision, or mobility require enhancements to existing computer-based systems in order to use these resources effectively. These laws address the requirement that acquisition and management of FIPS resources be conducted in a manner that ensures employees with disabilities access to computer and telecommunications products and services. The implementing regulations for these laws are contained in the Federal Information Resource Management Regulation (FIRMR), 41 CFR Chapter 201.

The interface designer must identify computer and telecommunication accessibility requirements for current and prospective users. The functional aspects of user requirements are an important part of system design and implementation. When automated information environments offer the needed flexibility, users with limitations in vision, hearing, or mobility are ensured full access and integration at a level equivalent to users without disabilities. Flexibility can be achieved in most information environments through off-the-shelf "drop in" or "add on" hardware and software enhancements that modify the common input (e.g., keyboard or mouse) or output (e.g., monitor or printer) interactions associated with computer operations. In addition to being more user-responsive, input capability may need to offer portability, speech input, or wireless connection to the computer. The output may need to be enhanced by magnified text or synthesized speech.

3.3.1 Visual I/O

3.3.1.1 Low Vision

The term "low vision" covers a broad range of possible conditions and types of visual impairment. The following techniques will enhance conditions for the visually impaired user.

- Use glare protection technology to minimize visual fatigue associated with glare on a monitor.

- Use monitors from 19 to 25 inches to allow increased character size and provide a larger image display.

- Use software or hardware to present images on a computer monitor in a large format. Character sizes can be increased.

- Use software to modify the print size on graphic printers.

- Allow the user to select color schemes for aspects of the application that do not involve coding or status. Ensure that a default scheme is easily available to restore the interface for subsequent users. The control of color will allow the user better contrast control.

- Code the keyboard tactilely with raised dot or bleb. Keycap labels with larger letters can be added.

- Allow the user to select font styles. Ensure that a default scheme is easily available to restore the interface for subsequent users.

3.3.1.2 Blind

The user with very limited or no usable vision will require additional interface enhancement.

- Provide an interface for the visually impaired by using speech recognition technology for input and speech synthesizers for output.

- Hardware and software are available to convert standard word-processing documents so they can be printed on a Braille printer. Dynamically displayed Braille is available on output devices. Braille note-takers' devices are available that are capable of Braille input and output to a personal computer (PC).

- The use of an optical character recognition (OCR) device can translate printed material to speech or Braille formats.

3.3.2 Hearing I/O

3.3.2.1 Visual Redundancy

Ensure that information conveyed by beeps or speech during computer-related tasks is also displayed visually for the user unable to benefit from the auditory information.

3.3.2.2 Amplification

Ensure that auditory output from the computer interface has adjustable volume and frequency range.

3.3.2.3 Signaling

Ensure that alerts and other signals related to the interface are presented in another modality, such as tactile or visual.

3.3.3 Mobility I/O

In addition to the computer interface, review the entire work environment for barriers to access. A variety of interface solutions are available for users with various degrees of limited mobility.

3.3.3.1 Keystroke Input

Modify the interface hardware and/or software to allow for sequential rather than simultaneous keystrokes. Create keyboard macros to reduce the number of keystrokes required. Adjust the repeat rate of keys, and modify to user requirements the pressure required to activate keys.

3.3.3.2 Keyboards

Alternate keyboards are available that may be more easily used by mobility- impaired individuals. Devices that replace keyboards (e.g., muscle switches, optical pointers, sip and puff systems) are also available.

3.3.3.3 Speech I/O

Using speech recognition technology for input and speech synthesizers for output will provide an interface for the mobility-impaired user.

3.3.3.4 Pointing Devices

The interface should not be dependent on pointing devices and should have redundant input/output capability through the keyboard. The selection of pointing device should consider the mobility parameters of the intended user.

3.3.3.5 Optical Character Recognition (OCR)

Using an OCR device can allow the translation of printed material to a speech-compatible interface.

This page intentionally left blank.

REFERENCES

Paragraph	References
3.1	DISA/CIM (1992)
3.2	Benson and Farrell (1988); Lloyd et al. (1991); Dye and Snyder (1991); Decker et al. (1991); Reger et al. (1989); Biberman and Tsou (1991)
3.2.1	Tannas (1985) page 11; Goode (1991); Beaton and Knox (1987)
3.2.1.1	Reger et al. (1989); Petrun et al. (1985)
3.2.1.1a	Laycock (1985)
3.2.1.1b	DoD (1989a) par. 5.2.6.8.3
3.2.1.1c	ISO (1991) Part 3 Addendum
3.2.1.1d	ISO (1991) Part 3 Addendum
3.2.1.2	ISO (1991) Part 3 Addendum
3.2.1.3	ISO (1991) Part 3 Addendum
3.2.1.4	Lloyd et al. (1991)
3.2.1.4a	Dye and Snyder (1991); Lloyd et al. (1991)
3.2.1.4b	Lloyd et al. (1991); Laycock (1985)
3.2.1.4c	Lloyd et al. (1991)
3.2.1.4d	Dye and Snyder (1991); Lloyd et al. (1991)
3.2.1.4e	Lloyd et al. (1991)
3.2.2	Snyder (1980); Goode (1991)
3.2.2.1	Petrun et al. (1985); Payne (1983)
3.2.2.1a	Payne (1983); Muracka et al. (1989)
3.2.2.1b	Biberman and Tsou (1991); Reger et al. (1989); Cristensen et al. (1985)
3.2.2.2	Kubota et al. (1988); Kubota Murushige Takabayashi and Kobayashi (1986)
3.2.2.3	Payne (1983)
3.2.2.4	Payne (1983); Muracka et al. (1989)

REFERENCES (cont'd)

Paragraph	References
3.2.3	Tannas (1985) Chapter 10; Goode (1991); Snyder (1980)
3.2.4	Snyder (1980); Goode (1991)
3.2.5.1	Reger et al. (1989); Petrun et al.(1985)
3.2.5.2	Morse (1985)
3.2.5.3	ISO (1991) Part 3
3.2.5.4	Hunter (1988)
3.2.5.5	Woodson et al. (1992)
3.2.6	Breen (1987); Breen (1989); Hurley and Hoffberg (1991); Carbone and MacIver (1987)
3.2.6.1	McNeese and Katz (1986); Shurtleff et al. (1982)
3.2.6.2	Woodson et al. (1992)
3.2.6.3	Blaha (1990); Breen (1989)
3.2.6.4	Shurtleff et al. (1982)
3.2.6.5	Woodson et al. (1992)
3.2.6.6	Blaha (1990)
3.2.6.6b	McNeese and Katz (1986)
3.2.6.7a	Woodson et al. (1992)
3.2.6.7b	Woodson et al. (1992)
3.2.6.8a	Woodson et al. (1992)
3.2.6.8b	Woodson et al. (1992); Vance (1987)
3.2.6.8c	Woodson et al. (1992)
3.2.7	Veron et al. (1990); Yeh and Silverstein (1989); Reinhart (1990); Beaton (1990); Snyder (1984); Williams (1990)
3.2.7.1	Beaton (1990)
3.2.7.2	Beaton (1990)
3.2.7.3	Yeh and Silverstein (1991)
3.2.7.4	Yeh and Silverstein (1991)

REFERENCES (cont'd)

Paragraph	References
3.2.7.5	Yeh and Silverstein (1989); Yeh and Silverstein (1991)
3.2.7.6	Yeh and Silverstein (1991)
3.2.7.6	Yeh and Silverstein (1991)
3.2.7.7	Yeh and Silverstein (1991)
3.2.7.8	Yeh and Silverstein (1991)
3.2.7.9	Yeh and Silverstein (1991)
3.2.8	Dominessy (1989)
3.2.8.1	Baggen et al. (1988)
3.2.8.2	Baggen et al. (1988)
3.3	GSA (1991); 41 CFR, Chapter 201

This page intentionally left blank.

4.0 SCREEN DESIGN

Screen design refers to the way information is arranged and presented on a display screen. It is difficult to develop a complete set of standard screen design guidelines for the variety of DoD systems, primarily because of the diversity of tasks being performed by users. Screen design requirements can vary extensively, depending on the function being performed by the system. Some systems, such as information management systems that rely heavily on databases, do not usually require immediate user response to information displayed on their screens. On the other hand, real-time tactical command and control systems require the user to make immediate decisions and to input commands based on information on the display screen. Screen design requirements are unique for each system, depending on the system's primary function. The designer needs to understand the primary function of the system being developed to provide an effective screen design.

The designer should also incorporate the following general principles of Human Factors Engineering (HFE) design into the screen design, regardless of the system function:

- Guide the organization of information by Gestalt principles of perception, such as rules of:
  - **Proximity**. The human perception system tries to organize objects into groups if they are near each other in space.
  - **Similarity**. Objects are perceived as a group or set if they visually share common properties, such as size, color, orientation in space, or brightness.
  - **Closure**. The human visual perception system tries to complete the figure and establish meaningful wholes. The incomplete object or symbol is seen as complete or whole.
  - **Balance**. Humans prefer stability in the perceived visual environment. The presentation of materials at right angles and in vertical or horizontal groupings is easier to look at than curved or angled visual images.
- Design display formats to provide optimum transfer of information to the user by the use of information:
  - **Coding**. Coding is the assignment of meaning to an arbitrary visual cue. Examples of information coding include the use of color coding of friendly/hostile threat, editable/noneditable text fields, or shape coding of map symbols such as bridges/towns/roads/terrain, or font coding of mandatory/optional text fields, or combinations of these coding methods.
  - **Density**. Density is the percentage of character positions on the entire screen that contain data (Galitz 1993). It is recommended that screen data density not exceed 30 percent.

  - **Grouping**. The general principle is that related information should be grouped together, but large groups of information should be broken up into subgroups. Related information is determined by what tasks are required to be performed by the user and by the users' perceptions of the information requirements.
  - **Enumerating**. The presentation of information in numerical or alphabetic or chronological order.

- Present information simply and in a well-organized manner.

- Improve user performance by implementing the following screen features:
  - An orderly, clutter-free appearance
  - Information present in expected locations
  - Plain, simple language
  - A simple way to move through the system
  - A clear indication of interrelationships.

- Design display formats to group data items on the basis of some logical principle, considering trade-offs derived from task analysis.

- Design screens to minimize pointer and eye movement requirements within the overall design. The goal to minimize eye and pointer movement must be considered within general task considerations, with logical trade-offs taken into account.

The information presented in Subsections 4.2 and 4.3 represent design considerations that should be applied to the screen design of all DoD systems. The information presented in Subsection 4.1 can be applied to all DoD systems, but is primarily concerned with general security (GENSER) interfaces that are used to work with or display classified material.

4.1 INITIAL SCREEN DESIGN FOR ACCESS-CONTROLLED WORKSTATIONS

This subsection provides guidelines for log-on, log-off, initial screen display, and management of access-controlled workstation resources. Although the focus is drawn from intelligence applications, the information presented applies to all system designs that display classified material and/or that control user access. The specific security requirements of the applicable domain relating to screen design must also be applied to any given application. This subsection applies primarily to GENSER requirements.

General principles that should be followed are that the system should provide both the necessary protection and be easy for the operator to use. The log-on for a system should not discourage use of the system by authorized users. Use of system resources and functions should be obvious and straightforward. Log-off should protect the data and preserve the information needed by the user without complicated and time-consuming procedures. Details for HCI design of CMW are contained in Appendix A.

4.1.1 Workstation Log-On

Develop a standard workstation log-on screen for each system (see Figure 4-1). All workstations should implement a screen saver, rather than continuously display a log-on screen or other display on an idle workstation. When the workstation has been idle for a maximum of 5 minutes, a screen saver should be activated and deactivated when new activity is initiated. When appropriate, allow the user to set the screen saver activation time to less than 5 minutes.

Guidelines for developing a workstation log-on procedure are as follows:

- Ensure that security authentication information (when required) is a combination of name, password, and/or other identification information required before a user can access system resources.

- Ensure that each prompt for the user's name, password, etc. is clearly labeled and displayed on a separate line.

- Display error messages clearly on the computer screen along with guidance on how to correct the error. Error messages or help generated during the log-on sequence should not convey information that could assist someone in breaking into the system.

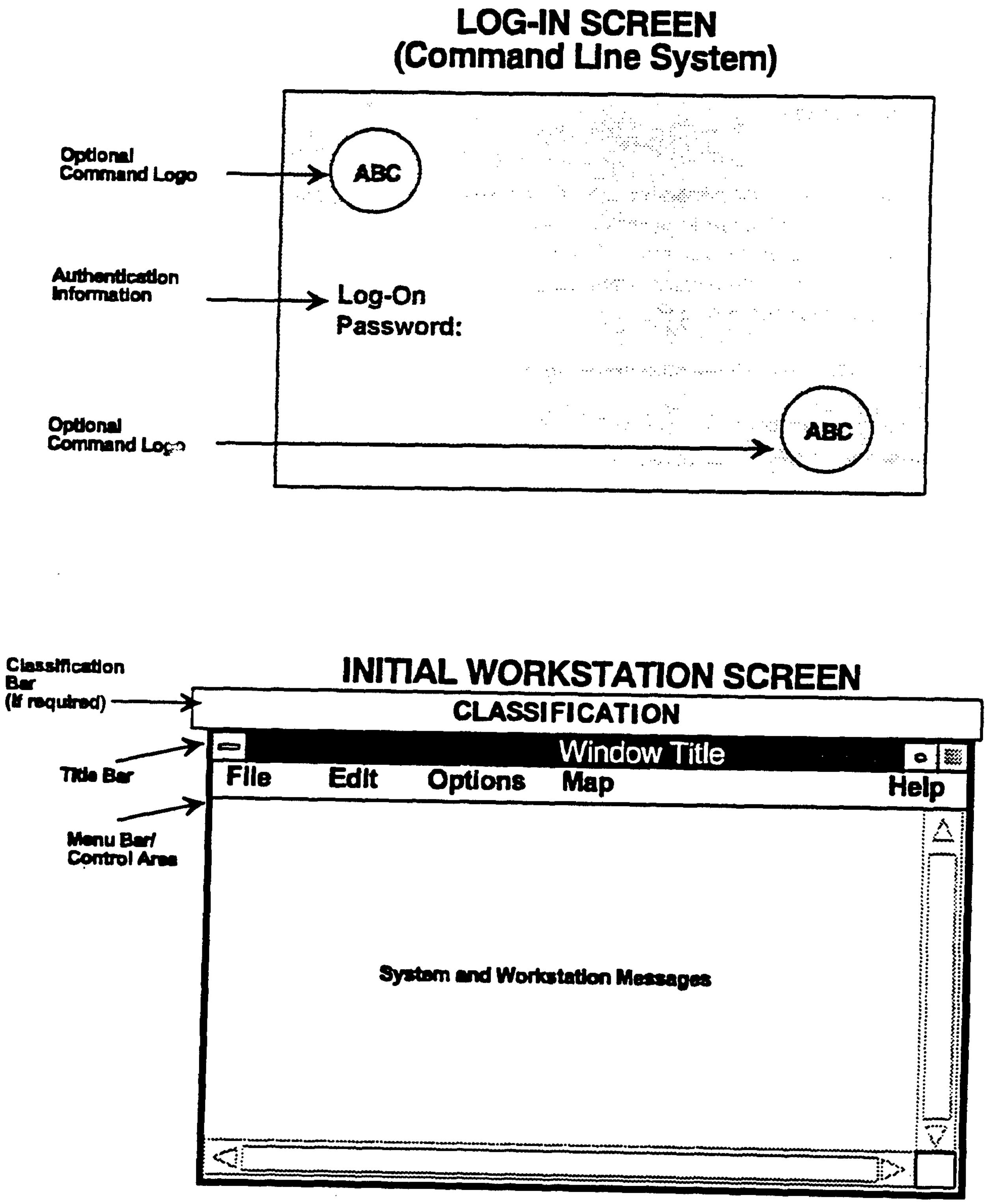
- When displaying a machine classification on a workstation accredited for system high operations, display the system high banner. The banner should be displayed in the color appropriate to the security level (see Paragraph 4.1.6).

4.1.2 Application Log-On

A primary DoD architectural objective for secure systems is to implement unitary log-on, but some systems will be unable to support this feature immediately. In systems where unitary log-on is not supported, many applications will require a separate authentication process before they can be accessed. Following selection of such an application by the user, display an additional log-on to prompt the user for the required authentication information.

4.1.3 Application Log-Off

Select the Exit function to accomplish application log-off. If work has not been saved, request that user confirm the quit, save modified data, or cancel the request. Application log-off exits an application and closes all windows associated with the application.


LOG-IN SCREEN
(Command Line System)

INITIAL WORKSTATION SCREEN

Figure 4-1. Sample Workstation Screens

4.1.4 Workstation Log-Off

Workstation log-off ends the session, closes all application windows, and returns the computer screen to the initial workstation log-on screen. If any applications are running, workstation log-off should also initiate an exit from all active applications. Workstation log-off requires user confirmation and is accomplished by selecting a log-off option from the resource manager window.

4.1.5 Initial Workstation Screen Display

When the user successfully completes the workstation log-on procedures and has been granted access to system resources, the initial screen should give the user access to the allowed system resources. Access can be accomplished by menus, icons, or interface structures, such as icon/tool bars. Some domains have specific requirements, such as the compartmentalized workstation requirement that a resource manager window will be displayed on the initial screen. Some tactical systems (especially sensor displays) will need to maximize available screen display space and will therefore design to minimize the space used by all other interfaces, including resource management interfaces. The designer of tactical systems may need to provide a resource management interface, such as an icon/tool bar with a toggle on/off option, to provide more user control of the screen display area. Resource management functions should be easily available to the user with a minimum of required keystrokes. The following are recommended basic resource management capabilities. General window functions are discussed in Section 5.0.

4.1.5.1 Resource Management

Resource management is the collection of functions that provides access to workstation resources and utilities (e.g., drives, printer, files, applications, software packages, etc.). This function is sometimes referred to as session management, but to avoid ambiguity, this *Style Guide* discusses the management of workstation resources under the generic term of resource management. The availability and display of resource capabilities should be determined by task requirements of the user. The following list of resource management capabilities provides examples of recommended functions:

- Program accesses
- Window snapshots (print screen)
- Access to common applications (e.g., word processor, spreadsheet)
- User preference/customization (e.g., left or right-handed mouse, color)
- Utilities (e.g., calculator, calendar, clock/alarm, note pad, mail)
- Display of system and workstation messages (error and status)

- End session/log user out of account
- Work file maintenance
- System-level help
- Security functions for authorized persons
- Device management capability (i.e., printer, mouse, facsimile [fax], etc.).

The resource management interface should present only those functions and applications a particular user is allowed to access. For example, only users authorized to perform certain security functions should have those options available within a resource management menu. Users may require data from several systems to perform their specific jobs. When multiple data sets must be accessed to satisfy a user query, it is the responsibility of the application to determine where the data reside.

4.1.5.2 Resource Management Interface

A resource management interface should contain, as a minimum, easy access for applications the user is authorized to use, including HELP. Workstations that require the display of system classification continuously can easily accommodate menus and/or icon/tool bar interfaces below the status display. Figure 4-1 illustrated a sample initial workstation screen. A long-term DoD objective is to implement user-oriented (e.g., help, messaging) resource management interfaces.

4.1.6 Classification Color Selection

The military intelligence community requires the colors listed below; the military community should follow these color selections. Classification bar color codes are shown as follows with their associated meanings.

Bar Color	**Meaning**
Green	Unclassified
Blue	Confidential
Red	Secret
Orange	Top Secret
Yellow	Sensitive Compartmented Information

When Sensitive Compartmented Information is displayed, both the classification (Secret or Top Secret) and "Sensitive Compartmented Information" must be displayed in the classification bar. The classification bar should contain two colors, orange or red, as appropriate, and yellow. If a two-color classification bar is unfeasible, a yellow classification bar should be displayed in which the classification and "Sensitive Compartmented Information" is displayed.

4.2 SCREEN DESIGN GUIDELINES

The visual design of the interface has increased in importance with the broad adoption of GUI as a standard interface. The screen design should be pleasing to the user, with screen elements arranged to be visually, conceptually, and linguistically clear and understandable. The visual presentation should be compatible with user expectations, using familiar concepts, terminology, and work flow. The concept of an easily learned and understood interface is central to the creation of screen designs. The interface should have the flexibility to support the requirements of users ranging from novice to expert. Users will be more productive if they control the interaction with the application, and this approach is recommended where possible. The best designs are easy to configure and reconfigure. One reason the use of prototypes for user input and feedback is recommended is because it will enhance the visual design process.

The functional design of the screen must deal with compatibility between the design and tasks to be performed. The user must be able to perform required tasks in a direct and obvious manner. Task flow should be predictable by the user and easy to follow. The functional layout should be efficient for the user, minimizing keystrokes and hand/eye movements where possible without impacting functional effectiveness. The functional interface should deal with errors and mistakes in a manner that allows easy correction and recovery. Responsiveness of the interface is important to screen design, together with consistency of the design. The most effective designs allow the user to rely on an interface that remains consistent throughout all system screens. The basic principle of functional screen design is to keep the interface as simple as possible and provide all the functionality required by the user to do the job.

The interface must conform to standards, guidelines, and requirements. The domain to which the system belongs, the functional system requirements, and the hardware platform selected will define many of the standards, guidelines, and requirements for the system. The screen interface should make hardware system-specific operations as transparent to the user as possible. The designer should review the standards, guidelines, and requirements before starting the design process. The design process should allow for trade-off in design to accommodate user needs and provide a means for user feedback into the design trade-off process.

4.2.1 Visual Design

The screen should be visually pleasing to the user. Using size, shape, location, and color can be counterproductive if these features startle or surprise the user. The goal of good visual design is computer interface design that visually encourages work flow, is easy to look at and easy to use.

4.2.1.1 Consistent Display Structure

Create display formats with a consistent structure evident to the user, so that display features are always presented in the same way.

4.2.1.2 Consistent Fields

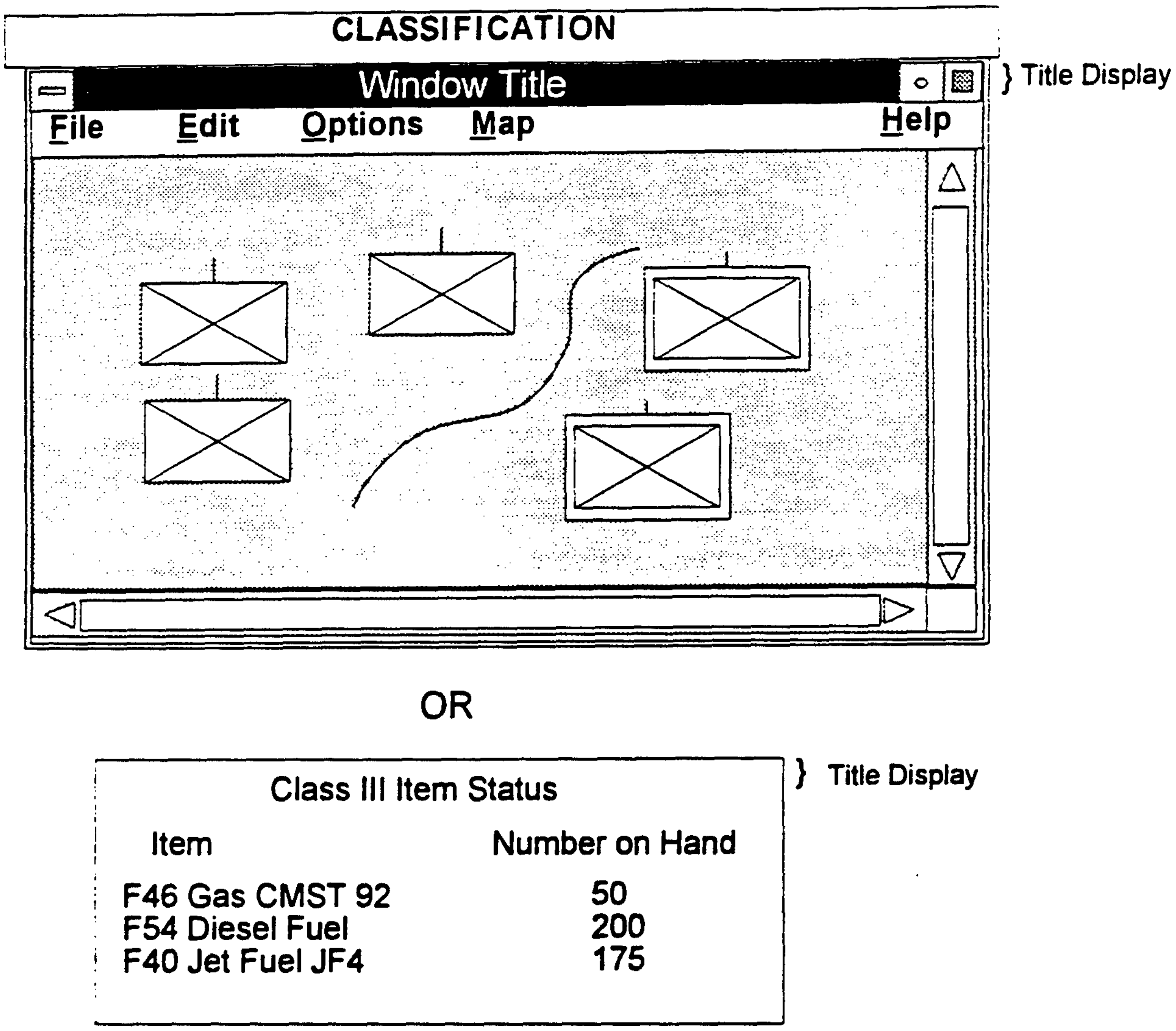
Use fields, such as data element names, group captions/titles, or window titles, that are consistently located and remain the same for each presentation.

4.2.1.3 General Format

- Make the different elements in a display format distinctive.
- Organize information on a display screen such that visual competition among distinct items of information is minimized.
- Use contrasting features, such as inverse video and color, to call attention to different screen components and urgent items.
- Arrange screen elements to be visually, conceptually, and linguistically clear and understandable.

4.2.1.4 Screen Organization

- Ensure that the order of data follows some principle that can be recognized and applied by the user.
- Begin every display with a title or header located at the top of the page or window, briefly describing display contents or purpose, as in Figure 4-1. In the special case of a CWS, a security banner must be the top-level label.
- Ensure that the area set aside for displaying messages is consistent. Text systems typically reserve the last few lines at the bottom of displays for status and error messages, prompts, and command entry, when appropriate (see Figure 4-2). This area is also used for a supporting data menu bar, including such items as a user note pad.
- For text displays, ensure that screen or focus window density (i.e., ratio of characters to blank spaces) does not exceed 60 percent of available character spaces. The data or information density (i.e., ratio of data characters to total display space) should not exceed 30 percent of the total screen or window. In this case, window size is defined as the default display size, not the maximum or minimum window size. For example:
  - A focus window 20 characters wide and 5 lines high has a total character space of 100 characters. The following sentence: *"The quick brown fox jumped over the large black dog."* uses 43 character spaces or 43 percent of the available 5-line display. If the following sentence is added: *"The information can be overloaded fast!"*, the total character space used is 77 characters or 77 percent of the total space. Thus, adding the second sentence creates too high a density for the size of the focus window.



Class III Item Status

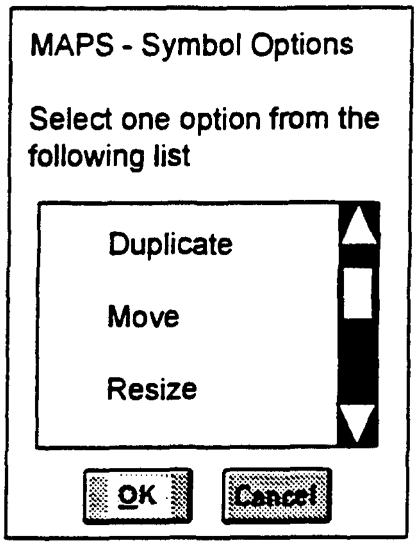
Item	Number on Hand
F46 Gas CMST 92	50
F54 Diesel Fuel	200
F40 Jet Fuel JF4	175

Figure 4-2. Example of How Specific Types of Information Should Be Located on a Window

- Calculating density is based on improving readability of displayed text. The display area does not include controls that may be within the window border. Controls are normally located in an area that is not made available to display the textual data.

- The principle of screen density applies to the display of graphic figures, but no standards for density maximums for graphic displays are available. The developer would be advised to test/prototype screen density designs with the user community.

- Highlight the instructions on how to use a screen or window at the top of the text, preceding response options, as illustrated in Figure 4-3. Include instructions on the disposition of a completed screen or window at the bottom.



MAPS - Symbol Options

Select one option from the following list

Figure 4-3. Example of the Proper Location of Display Screen Instructions

- Assign functional fields for particular kinds of data, such as program messages, error messages, system messages, and alarms. These fields and displays should be consistently located in the physical space of the screen or window.

4.2.1.5 Primary Viewing Area

When data and terms are particularly important, require immediate user response or, when they are more frequently displayed, group them in the primary viewing area of the user.

4.2.1.6 Arrangement of Data on Screen

Arrange and group data on application display screens to differentiate between instructions and data and to facilitate observation of similarities, differences, and trends for the most common uses.

4.2.1.7 Cohesive Groupings

Provide cohesive groupings of screen elements by using blank space, surrounding lines, different intensity levels, etc.

4.2.1.8 User Attention

Techniques that direct user attention should be used carefully or they will lose their effectiveness. A number of visual techniques may be used to attract the attention of the user.

The following list contains visual approaches, their properties and parameters that may be used to attract user attention:

- **Intensity**: Do not use more than two levels.
- **Marking**: Underline, arrows, bullet, dash, asterisk.
- **Size**: The maximum number of sizes should be four or less.
- **Blinking**: Blink rates should be in the 2 to 4 hertz range.
- **Choice of Fonts**: The number of fonts should be three or less.
- **Inverse Video**: Use inverse coloring.
- **Color**: Use up to four standard colors.

4.2.1.9 User Feedback (Prototyping)

The visual screen design process should include prototype or sample screen presentations to users. Sampling user opinion and obtaining user input to the visual design process is critical to developing an effective interface for a system. The basic principles of screen design require that user feedback be an integral part of the development process.

4.2.2 Functional Screen Design

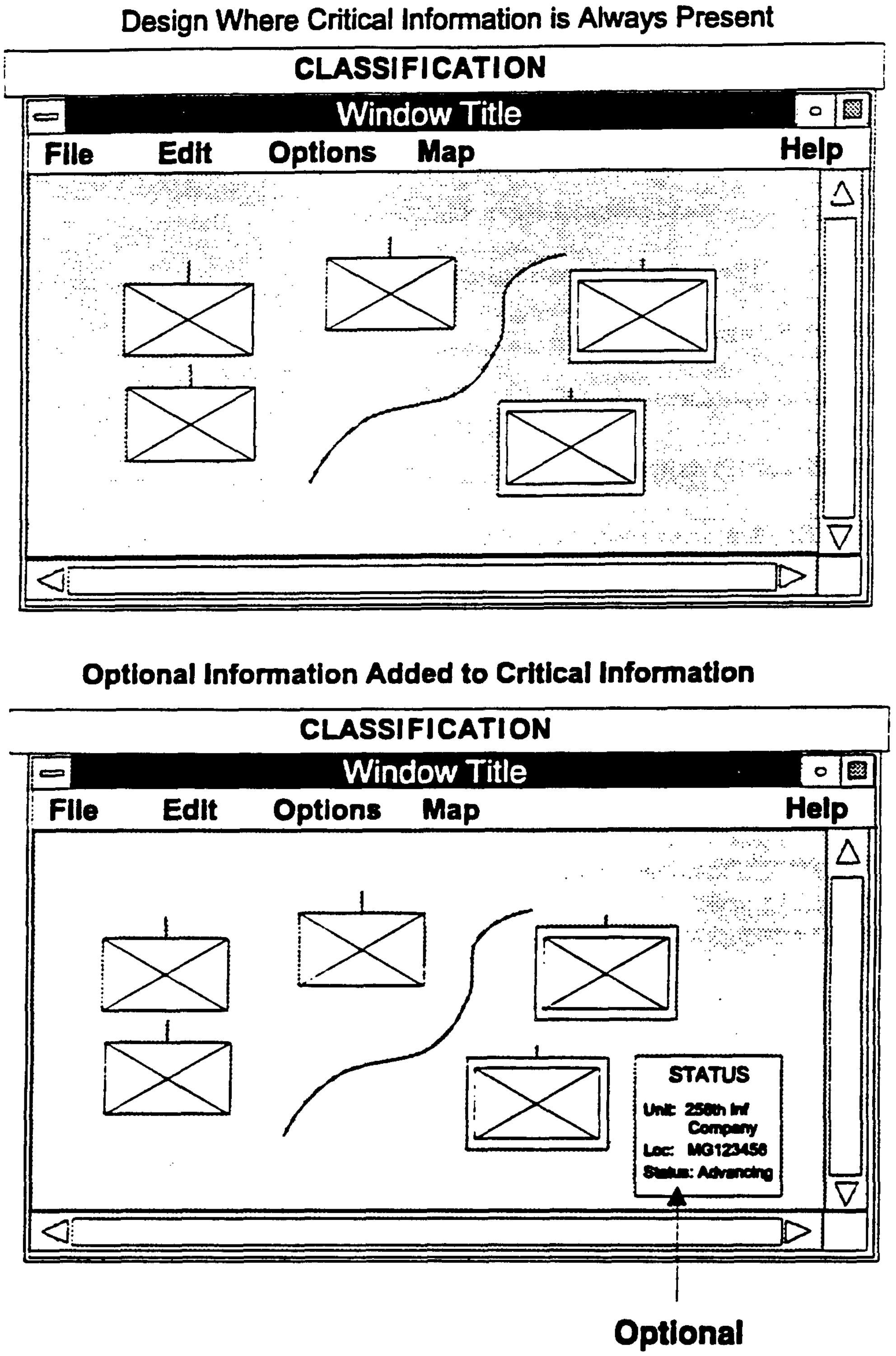
The functional screen design integrates the user task requirements with the available computer functionality to optimize the user's performance. The interface must present an obvious and predictable work flow that is efficient for the user. The interface should be as simple as possible and still give the user the functionality to complete tasks effectively.

4.2.2.1 Information Display Based on Criticality

The screen design procedures for a system should establish a set of criteria for prioritizing different levels of displayed information. For example, in military tactical systems, critical tactical information should always be displayed, whereas optional information should be available by request. See Figure 4-4 for an illustration of this principle, using a military land-based situation map overlay. The position of the units is critical information, whereas details of the unit are available on a pop-up box.

4.2.2.2 Display Only Critical Information

Minimize the text information density on a system display by presenting only information essential to the user at the time. High density of information on a screen or window results in the user needing extra time and making errors in finding information. It is recommended that screen or window information density be less than 30 percent. Information is variable data and should be distinguished from labels, titles, or lines/boxes. The density of graphic displays should also be minimized, using input from end users as guidance.



Design Where Critical Information is Always Present

Optional Information Added to Critical Information

STATUS

Unit: 256th Inf Company

Loc: MG123456

Status: Advancing

Figure 4-4. Tactical Information Display Options

4.2.2.3 Integrated Display

When the user needs specific data displayed concurrently to judge a time-critical task (e.g., a military tactical situation), provide those data in an integrated display rather than partitioning them into separate windows. Using integrated data displays will facilitate decision-making and time-constrained tasks.

4.2.2.4 Information Format

Present information in a directly usable form. Do not require the user to decode or interpret data. The structure of information presented on the screen should be consistent. This helps the user develop a perceptual model of the interface.

4.2.2.5 Grouping for Data Comparison

If users must analyze sets of data to discern similarities, differences, trends, and relationships, structure display formats so the data are grouped consistently.

4.2.2.6 Important Data Placement

Where displayed data are used in some spatial or temporal order, consider grouping those data by sequence of use to preserve that order.

4.2.2.7 Efficient Layout

A design goal for DoD is to minimize keystrokes and hand/eye movements, but this goal must be implemented carefully. The designer should consider functional requirements and trade-offs among task requirements, while striving for the goal of minimum keystrokes and hand/eye movements. An efficient layout will incorporate consideration for function as well as interaction.

4.2.2.8 Error Management

The screen design must include error management. The most effective error management is an interface design that minimizes errors. Errors, in fact, will occur, and the functional interface must include provision for managing errors and mistakes related to functional tasks performed by the user. The interface should allow easy correction and recovery while protecting the application from catastrophic user errors. See Subsection 8.2 ON-LINE HELP for more information.

4.2.2.9 Functional Trade-Off

A functional trade-off analysis for the system under development should be performed to determine the tasks best performed by automation and the tasks best performed by humans. Automation can be an efficient partner by handling routine tasks, while reducing the impact of tedious and error-prone tasks. The functional screen design must assign the task to the most appropriate resource, either computer or human. Tasks should not be automated for the sake of automation, but the user should not be burdened with tasks better done by automation. See the list below, which is based on a table from Shneiderman (1992, page 84).

Tasks Best Performed By Humans	Tasks Best Performed By Automation
- Remember principles and strategies	- Recall quantities of detailed information
- Retrieve pertinent details without a prior connection	- Process quantitative data in prespecified ways
- Adaptability	- Accuracy
- Reason inductively - generalize from observations	- Reason deductively - infer from a general principle
- Sense unusual and unexpected events	- Monitor prespecified events
- Act in unanticipated emergencies and novel situations	- Perform repetitive preprogrammed actions reliably
- Draw on experience and adapt decisions to situation	- Perform several activities simultaneously
- Detect stimuli in noisy background	- Calculate accurately and quickly

4.2.3 Screen Design Standards, Guidelines, and Requirements

4.2.3.1 Format

- Use abbreviations appropriately and consistently. Provide a key or built-in reference table. Abbreviations should conform to standards (e.g., AR310-50 [U.S. Department of the Army 1985a], MIL-STD-12D [DoD 1981], MIL-STD-411E [DoD 1991], and MIL-STD-783D [DoD 1984]). The domain-level style guide should cite the domain-selected standard for abbreviations. Do not place periods after abbreviations. Applications requiring extensive text input should provide an on-line spell-checker that addresses abbreviations and acronyms.

- Use short, simple statements in text.

4.2.3.2 Data Organization

- Break large portions of text into smaller, meaningful groups to minimize the amount of information to be attended to at one time. See the examples below.

Poor:
The 3rd Bn is currently located at 32UNA100100, moving to contact in sector 8, with 80% strength, supported by an armor platoon.

Good:
3 Bn Status:

- *At 32UNA100100*
- *Moving to Contact in Sector 8*
- *80% Strength*
- *Supported by an armor platoon*

- Use blank space to structure a display.
- For screens containing large amounts of text, consider using two columns of text to improve readability.
- Ensure labels are sufficiently close to their related data fields but separated by at least one space.
- Provide adequate spacing between words and lines of text for better legibility. Separate paragraphs with a blank line.
- Present a series of data elements vertically, not horizontally, in text, as follows.

Vertical - Easy to Read

Class I	Class II	Class III
Item 1 Item 2 Item 3	Item 1 Item 2 Item 3	Item 1 Item 2 Item 3

Horizontal - Difficult to Read

Class I	Item 1	Item 2	Item 3
Class II	Item 1	Item 2	Item 3
Class III	Item 1	Item 2	Item 3

- Provide an obvious starting point for information.
- Justify columns, as noted and illustrated below.
  - Left-justify alphanumeric columns to permit rapid scanning.
  - Right-justify numerical data without decimals.
  - Justify numerical data with decimal points by the decimal.

Poor	Good
Washington DC	Washington DC
Cars	Cars
People	People
400	400
4210	4210
39111	39111
1.5	1.5
10.36	10.36
1.365	1.365

4.2.3.3 Line Organization

- On a full-screen text display, use only 70 character positions on the standard 80-character line to increase reading efficiency. This is most important when detailed reading is the user's primary task. When displaying text in windows format, where possible, ensure word wrap to prevent excessive scrolling side-to-side. Text presented in a window should have blank space at the start and end of each line to increase readability of the text. It is recommended that the blank space be approximately the size of two average text widths.

- Display no more than 35 to 40 characters per column on each line for information presented in columns.

4.2.3.4 Character Design

- Use capital letters for typographic coding, headlines, and where special emphasis is required, such as some captions and labels. However, it is usually best to capitalize only the first letter, for example, in a horizontal series such as button labels.

- Do not use all capital letters in running text or tables, as this impairs word recognition, reduces readability, and limits space between text lines.

- Ensure spacing between characters (both fixed and proportional width) at 20 to 50 percent of character height. Spacing between lines should be equal to character height (see Figure 4-5).

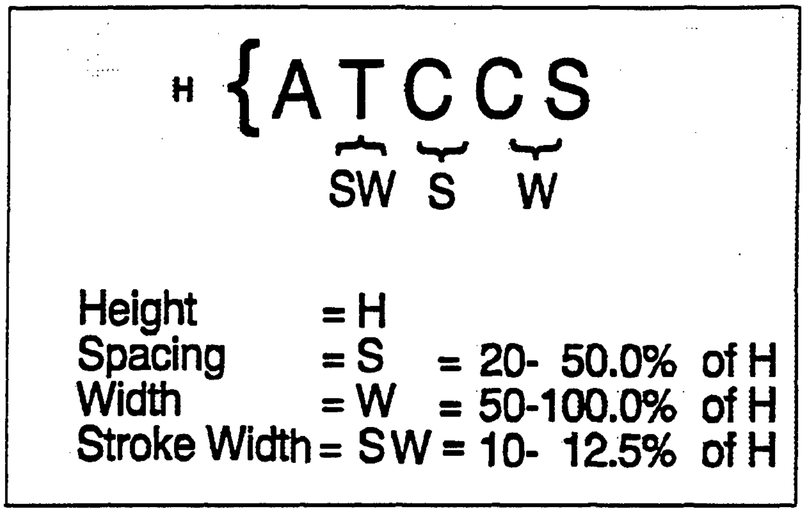

H
ATCCS
SW
S
W
Height = H
Spacing = S = 20- 50.0% of H
Width = W = 50-100.0% of H
Stroke Width = SW = 10- 12.5% of H

Figure 4-5. Example of a Character Size and Spacing

- Ensure that the minimum height of displayed characters is 1/200 of viewing distance (approximately 17 minutes of visual angle). Therefore, a viewing distance of 36 inches requires 0.18-inch character height on the display screen. Character width should be 50 to 100 percent of character height. Character stroke width minimum is 10 to 12.5 percent of character height. Maximum text size should not exceed 10 percent of the available vertical display area on a full-size screen.

- Ensure that characters contain a minimum 7 x 9 dot matrix construction for better readability

- Some applications require over-the-shoulder reading of characters on the screen and should be legible to a person standing behind the user (e.g., operations in a tactical environment). Ensure that the screen viewing distance referred to in Paragraph 4.2.3.4d reflects the anticipated maximum viewing distance. Large fonts with broad stroke widths are recommended to improve readability. Selections of background color and contrasting foreground (text) color should ensure sufficient contrast for good readability.

- The usual font size designation is given in points. Display of text fonts on screens is proportional to point size, but the actual size of displayed text is related to screen size and application software. Font point size only controls the actual size of printed output. It is recommended that screen text size be reviewed and adjusted in relation to the objective hardware system.

4.2.3.5 Scrolling of Data

- Display text information statically on the screen, rather than constantly scrolling it across the screen.

- If text is meant to be scanned by constant scrolling, columns with 35 characters or fewer per line are preferred.

4.2.3.6 Data Grouped Alphabetically or Chronologically

When no appropriate logic exists for grouping data by sequence, function, frequency, or importance, adopt some other principle, such as alphabetical or chronological grouping.

4.2.3.7 Paging Crowded Displays

When a window contains too much data for presentation in a single pane, normally the window has the ability to scroll the data. It is recommended that side-to-side scrolling be avoided where possible. If the data must be presented in block or groups, partition data into separately displayable window panes or pages. Refer to Section 5.0 WINDOWS.

4.2.3.8 Related Data on Same Page

When partitioning displays into multiple pages, take into account the type of data being partitioned, and display functionally related data items together on one page.

4.2.3.9 Multiple Pages Labeling

In a multipage display, label each page with a unique identifier that shows its relation to the other pages (e.g., page 1 of 5).

4.2.3.10 Screen Viewing Distance

The minimum viewing distance from the user's eye to the monitor must be equal to or greater than 30 centemeters (cm) or 12 inches. For best visual acuity, it is recommended that the viewing distance be set to the range of 30 to 40 cm (12 to 16 inches). Viewing distance must be considered when character and symbol sizes are determined for the screen design.

4.2.3.11 Text Readability

The readability of displayed text is maximized if the character height is in the range of 16 to 24 minutes of visual arc (min). The preferred size is 20 to 22 minutes of visual arc. The designer must determine the maximum viewing distance from the display, then calculate the minimum size of the text, using the formula:

$$\text{Visual Angle (min)} = \frac{(57.3)\,(60)L}{D}$$

where L = size of the object, and D = distance from the eye to the object.

The list on the following page provides examples of calculations based on the above formula. For example, to obtain a visual angle of 20°, when the user is 32 cm from the screen, this requires a screen object (font height) to be 0.186 cm.

4.2.3.12 Input Prompts

When command line interfaces are used, display the input prompt at a standard location, next to the command entry area of the display.

4.3 COLOR

The use of color as an information discriminator is crucial, especially as emerging command and control systems implement GUIs and high resolution color graphics displays. Color can be a very effective discriminator, for example, by decluttering a display and thus improving task performance. Color can also introduce the very clutter and performance degradation it attempts to reduce. For these reasons, color in a display must be used very carefully.

The individual or team responsible for screen design must be sensitive to the many factors that affect how a person perceives and reacts to color as an information discriminator. An in-depth discussion on visual perception, color, and human performance is beyond the scope of this document; nevertheless, basic information is needed. Definitions of key terms associated with this subject are provided.

Visual Angle	D = Distance to object (cm)	L = Height of object (cm)
17 min	30 cm	0.148 cm
17 min	31 cm	0.153 cm
17 min	32 cm	0.158 cm
17 min	33 cm	0.163 cm
17 min	34 cm	0.168 cm
18 min	30 cm	0.157 cm
18 min	31 cm	0.162 cm
18 min	32 cm	0.167 cm
18 min	33 cm	0.172 cm
18 min	34 cm	0.178 cm
19 min	30 cm	0.165 cm
19 min	31 cm	0.171 cm
19 min	32 cm	0.177 cm
19 min	33 cm	0.182 cm
19 min	34 cm	0.188 cm
20 min	30 cm	0.174 cm
20 min	31 cm	0.180 cm
20 min	32 cm	0.186 cm
20 min	33 cm	0.192 cm
20 min	34 cm	0.198 cm

Definitions of Key Terms for Color Usage

TERM	DEFINITION
Achromatic	Colorless; lights that have no definite hue, black and white are achromatic.
Brightness	The perceptual (psychological) correlate of intensity that ranges from dark to bright.
Chroma	The intensity or vividness of a color, its brightness or dullness.
Chromatic	Highly colored.
Discrimination	Degree to which a human visual system can sense differences in the physical characteristics of an image.
Hue	The psychological attribute of color sensation associated with the physical property of visible wavelengths. The name of a color such as blue, red, green, or orange.
Legibility	Ability to identify an alphanumeric character or symbol. A criterion of image quality.
Luminance	The amount of light reflected or emitted by a surface, measured in footLamberts.
Luminance	Ratio of the foreground brightness compared to the Contrast background brightness.
Monochromatic	Consisting of one color or hue.
Recognition	Ability to recognize or interpret the meaning or association of an image.
Saturation	The degree to which a hue differs from a gray of the same lightness.
Shade	The darkness of a hue produced by adding black.
Tint	The lightness of a hue produced by adding white.

The designer should recognize the following important guidelines for using color in computer display systems:

- Both brightness and type of lighting (e.g., incandescent versus fluorescent) can affect how colors are perceived. For example, bright ambient light desaturates display colors, leading to degraded color identification and discrimination. It may shift the eye's adaption, also reducing the ability to discriminate color. In essence, identically colored objects can be perceived as being dissimilar under different lighting conditions.

- How the color of an object is perceived is directly related to the color surrounding it.

- Visibility and readability are a direct result of the contrast between the foreground and the background.

- Use color sparingly as an information discriminator. Color rapidly loses meaning and, when overused, may impede rather than enhance human performance.

- Use colors consistently within a display and across a set of displays for an application.

- Ensure that the meaning of color is consistent with user expectation.

- When using color to impart a specific meaning to the user, use an additional, redundant form of coding, such as pattern or shape. This ensures that the correct meaning will be conveyed should the user have a color vision deficit, or should color be unavailable on the screen.

- Standardize color coding for operational applications. Although flexibility in color coding schemes may be desirable for a terminal dedicated to a single user, color coding should be standardized for operational applications. Because of the variety of users on a tactical terminal, a terminal with a uniquely customized color coding scheme may be very difficult to interpret. Allow the user to select color schemes for aspects of the application that do not involve coding or status, unless the primary task performed by the user requires the capability to manipulate these colors and making these changes will not negatively impact the performance of other users. A default scheme should be easily available to restore the interface for subsequent users.

- A requirement for adjustable colors is created because of portable applications among hardware configurations. Each hardware system display has different color perceptual values and color names. Portable applications must accommodate these differences. Assign status colors during installation; allow the user to adjust these colors only if essential to the task being performed with the system.

The following subsections provide more detailed guidelines for using color in DoD systems. Note that many guidelines contained in this subsection are based on the use of color in text-based software, primarily because the majority of past research in color usage was done with text applications. Although results of research with GUIs and graphical presentation tend to confirm these principles and new information is emerging, more research is needed on using color in tactical graphics applications, especially in foreground/background combinations for colored map graphic displays.

The designer should also note that it is easier to define color combinations to avoid than to identify a single best way to utilize color. Color choice and/or color combination tend to be a matter of personal preference. For example, high contrast between foreground text and background is crucial but can be accomplished with a number of color combinations. This ambiguity becomes all the greater when developing design guidance for the different command and control applications represented across military systems. The designer should utilize those domain-level guidelines most relevant to the particular application.

Use color when a basic monochromatic presentation of tactical information needs to be augmented for the user to gain a more effective understanding of the information being presented.

4.3.1 General

Reference to color can mean a family of color, such as reds; it can mean a Red, Green, Blue (RGB) definition, such as red 1,0,0; or it can mean a specific frequency of light. The *Style Guide* will use an underline for color names that refer to primary (i.e., full gun) RGB. Otherwise the reference to color is to a family or perceived color.

4.3.1.1 When to Use

Use color carefully (as a coding method, color can rapidly lose its effectiveness). When necessary, use color:

- To attach specific meaning to tactical information presented in the form of text or symbology
- To direct user's attention to the most important or time-critical information on the screen (i.e., information category headings, system and user errors, information requiring immediate attention, key data items, window titles)
- To enable a user to differentiate rapidly among several types of information, especially when the information is dispersed on the display
- To increase the amount of information portrayed on a graphic display by adding color in addition to shape
- To indicate changes in the status of graphical data.

4.3.1.2 Constraints on Use

The user with defective color vision will have difficulty discriminating among the colors. Color vision deficiency occurs in about 8 to 10 percent of the general male population and about 0.4 to 0.6 percent of the female population in the United States. Consider the following when including color in display screens:

- Add color coding only after displays have already been designed as effectively as possible in an achromatic format.

- Color only logically related information with similar hues. Consider spacing or highlighting instead of, or in addition to, color.

- When emphasizing tactical information by means of color, use a color for more important information that is brighter than adjacent color-coded information. Ensure the choice of colors is consistent with the user's expectations for the information being coded (see Paragraph 4.3.2).

- Do not use color coding when it might confuse users with defective color vision or when the use of color reduces screen readability. If color must be used, consider the following:

  - When the user must compare data, such as those contained in graphs based on color, the list below suggests combinations to use and avoid as comparison colors for application information requiring important or frequent discriminations. These combinations do not apply to text (foreground) and screen (background) color combinations.

Data Comparison Color Combinations

RECOMMENDED	**AVOID**
White/Green	Red/Blue
Gold/Cyan	Red/Green
Gold/Green	Red/Purple
Green/Magenta	Red/Yellow
Green/Lavender	Red/Magenta
Cyan/Red	White/Cyan
White/Gold/Green	White/Yellow
White/Gold/Blue	Blue/Green
White/Gold/Magenta	Blue/Purple
White/Red/Cyan	Green/Cyan
Red/Cyan/Gold	Cyan/Lavender
Cyan/Yellow/Lavender	Red/Yellow/Green
Gold/Magenta/Blue	Red/Blue/Green
Gold/Lavender/Green	Red/Magenta/Blue
	White/Cyan/Yellow
	Green/Cyan/Blue

- Especially when using color combinations from the "avoid" side of the preceding list, provide additional cues, such as brightness and saturation, to enhance discriminability.

- Do not code solely by color. Make color coding redundant with some other display feature, such as shape, pattern, or actual text content.

- Avoid requiring the user to discriminate between colors in small areas of the display. Small, color-coded areas are subject to loss and bleeding of colors. Use achromatic colors (i.e., black or white) if coding must be done in small areas.

- White is a good choice to highlight data that require particular attention, except on a display with white or near-white background. Do not use white excessively as a highlighter, as it can create a glaring brightness that may interfere with screen legibility. When status changes are signaled by color, do not use that color to highlight text. Signal any status changes using color coding by a ball or box next to the text.

- Ensure that contrast is high between the text or graphical object and its background, to enhance screen readability. Generally, the color foreground should differ from its background by a minimum of 100 E (Commission International d'Eclairage (CIE) Yu'v') distances, when luminance values range from 0 to 255. A luminance equation (*intensities have been normalized from 0.0 to 1.0) that can be used for contrast determination is:

  $Y = .30^{*}\underline{R}ed + .59^{*}\underline{G}reen + .11^{*}\underline{B}lue$

  The minimum normalized difference should be greater than 0.35 for good contrast. The luminance difference value for a number of standard colors is listed below.

- Based upon the tables listed, values **red** and **black** or **black** and **blue** should not be used, while **white** and **blue** or **black** and **cyan** have acceptable contrasts. Minimum luminance contrast ratios are required for specific tasks. For discrimination and legibility, acceptable ratios of foreground-to-background luminance contrast range from 6:1 to 10:1, or 1:6 to 1:10. The list below provides guidance for specific conditions. Using pure white or black as a background color is not recommended. Unsaturated hues provide the best background contrast. The use of high luminance backgrounds tends to cause user eye strain. Therefore, task requirements should be considered when selecting high or low background illumination.

NOTE: When calculating luminance ratios using black (i.e., zero luminance), use a value of one (1) to avoid dividing by zero (0). This is based on luminance values that range from 0 to 255.

Normalized Standard Color Luminance (Y)

	R	G	B	Y
Black	0.0	0.0	0.0	0.00
White	1.0	1.0	1.0	1.00
Red	1.0	0.0	0.0	0.30
Green	0.0	1.0	0.0	0.59
Blue	0.0	0.0	1.0	0.11
Cyan	0.0	1.0	1.0	0.70
Magenta	1.0	0.0	1.0	0.41
Orange	1.0	0.5	0.0	0.60
Yellow	1.0	1.0	0.0	0.89

Luminance Differences

	Black	White	Red	Green	Blue	Cyan	Magenta	Orange	Yellow
Black	XXX	1.00	0.30	0.59	0.11	0.70	0.41	0.60	0.89
White	——	XXX	0.70	0.41	0.89	0.30	0.59	0.41	0.11
Red	——	——	XXX	0.29	0.19	0.40	0.11	0.30	0.59
Grecn	——	——	——	XXX	0.48	0.11	0.18	0.01	0.30
Bluc	——	——	——	——	XXX	0.59	0.30	0.49	0.78
Cyan	——	——	——	——	——	XXX	0.29	0.11	0.19
Magenta	——	——	——	——	——	——	XXX	0.19	0.48
Orange	——	——	——	——	——	——	——	0.48	0.30

Recommended Luminance Contrast Ratios

CONDITION	RATIO OF FOREGROUND TO BACKGROUND
Bright Ambient Illumination	>7:1
To Attract Attention	>7:1
To Sharpen Edges	>7:1
Continuous Reading	3:1 to 5:1
Dark Ambient Illumination	3:1 to 5:1
Camouflage Images or Smooth Edges	<3:1

4.3.2 Color Selection

4.3.2.1 General

- When selecting colors for coding discrete categories of information displayed on a screen, ensure that those colors are easily discriminated in all expected operational environments.
- To aid in color discrimination, use colors that are as widely spaced along the visible color spectrum as possible. The following colors, listed by their wavelengths in millimicrons, are spaced widely enough for easy discrimination from one another.

Color Wavelengths in Millimicrons:

Red	700
Orange	600
Yellow	570
Yellow-green	535
Green	500
Blue-green	493
Blue	470

- Use an unobtrusive color to display information used infrequently on a screen. Unobtrusive colors have shorter wavelengths.

- Use warm colors (colors with a longer wavelength, such as red or orange) to convey action or the requirement for a response. Use cool colors (colors with a shorter wavelength, such as blue or green) to convey status or background information.

- Ensure that each color represents only one category of displayed data (i.e., those defined in Paragraph 4.3.2.6). A mismatch of color and color association slows recognition time and increases misidentification of words.

4.3.2.2 Consistency

- Apply color consistently from screen to screen and from application to application to ensure that the user can make the proper interpretations. This is applicable both within and across DoD systems. For example, do not use status colors as window borders unless status coding is intended.

- Color coding should be consistent with the interaction of the label's color and the color associations of the words in the label. For example, the word ENEMY, if color-coded, should be red rather than green.

- Choose colors for coding based on conventional associations with particular colors. These should conform, if possible, to those specified in the appropriate domain-level documents, such as *Army FM 101-5-1: Operational Terms and Symbols* (U.S. Department of the Army 1985b).

4.3.2.3 Number of Colors to Use

- Implement color coding conservatively, using relatively few colors to designate critical categories of displayed data and only where it will help user performance.

- Use no more than four colors at one time when using alphanumeric screens, with a maximum of seven total for all screens.

- Use four standard colors, reserving others for occasional use. Humans can easily discriminate only eight or nine highly saturated colors; the recommendation is *not* to exceed seven. Extensive coloring creates a brighter-than-necessary display, with subsequent negative impact on user performance.

4.3.2.4 Pairing of Colors

Colors should be carefully paired on a screen to maximize human performance.

- Avoid simultaneous use or close proximity of highly saturated, spectrally extreme color pairs on a display screen. Examples include such color pairs as red and blue, yellow and purple, or magenta and green. This creates an effect where one colored object will appear closer than another (a 3D effect called chromostereopsia). This effect is most significant with red and blue.

- To emphasize different tactical information in text and presentation graphics displays, the color choice rules may be broken and contrasting colors such as red and green or blue and yellow may be used. However, in color choice, be consistent with the guidance provided in other parts of this section.

- To convey similarity in tactical information in text and presentation graphics displays, use similar colors, such as orange and yellow or blue and violet.

- Avoid using extensive coloring (e.g., many different colors) for the background, segments of the background, or particular regions surrounding individual characters or symbols.

- Avoid using pairs of monosaturated primary colors, such as red-green or blue-green, because of the possibility that the user is partially or completely color-blind. The red-green should always have some blue tones, and the blue-greens always should have some red tones.

4.3.2.5 Color Selection and Ambient Illumination

The level of ambient illumination directly affects the perceived brightness and hue of a color. Consider the following when designing a color display:

- Use green, as it provides good general visibility over a broad range of intermediate luminances.

- Use red under high ambient lighting but not in low lighting.

- Use yellow, as it provides good general visibility over a broad range of luminances.

4.3.2.6 Specific Color Meanings

Use the colors and associated meanings listed below for designing military color coding. The exact color values selected are dependent on the background upon which they are to be displayed.

Color	Meaning
Green	Non-alert, neutral forces, forces or situation at acceptable condition, obstacles on map graphics, ON as opposed to OFF
Blue	Friendly forces symbology, cool, safe, nitrogen, deep
Red	Alert, forces or situation at critical condition, enemy symbology, stop, dangerous, oxygen, hot
Yellow	Forces or situation at marginal condition, unknown forces, caution, NBC areas on map graphics
Black	Political boundary, image or figure edge

4.3.2.7 Using Blue

Blue as a background color is most effective for tasks performed at close distances.

- Because the eye is relatively insensitive to blue, blue lines or dots will be very difficult to resolve. Avoid using saturated blue for small lines or dots when the background is dark.

- Use saturated blue only for background features in a display, not for critical data.

4.3.2.8 Use of Color Keys

While not recommended, there may be some circumstances where the system designer must allow the screen design to deviate from the color meanings provided in the previous lists on classification bar color codes and associated meanings, or use other colors. When this happens, it is important to include on the display a key that explains the color meaning.

- Ensure that the color key is readily accessible visually on the display without having to scroll or expand the screen or window.

- Ensure that the colors in the key have the same appearance as the color being defined.

4.3.2.9 Large-Screen Display Periphery Colors

Avoid the use of red and green in the periphery of a large-scale display. Yellow and blue are good periphery colors.

4.3.2.10 Color Sets

When selecting color sets for displays, ensure that contrast is high between foreground objects and background displays. Black provides high contrast with light shades or with white. No color should be contrasted with a lighter or darker shade of itself, if this can be avoided (e.g., it cannot be avoided on monochrome displays). The hypertext version of this subsection will include a visual comparison of background versus foreground selections.

4.3.3 Tonal Color Coding

4.3.3.1 Color Coding for Relative Values

When relative rather than absolute values of a variable are important, display gradual color changes of a single color as a tonal code to show the relative values of a single variable. Display a monochromatic shading rather than spectral codes (different colors).

4.3.3.2 Ordered Coding

If different map areas are coded by texture patterns or tonal variation, order the assigned code values such that darkest and lightest shades correspond to extreme values of the coded variable.

4.3.4 Color-Coded Symbols

Use the following guidelines with symbols that are color-coded.

4.3.4.1 Color-Coded Symbol Size

Ensure that color-coded symbols subtend a minimum of 20 minutes of visual arc. The designer must determine the maximum viewing distance from the display, then calculate the minimum size of the object, using the formula: Visual Angle (Min.) = $\frac{(57.3)\ (60)L}{D}$

where L = size of the object, and D = distance from the eye to the object.

The units of measure can be inches or centimeters (see Figure 4-6).

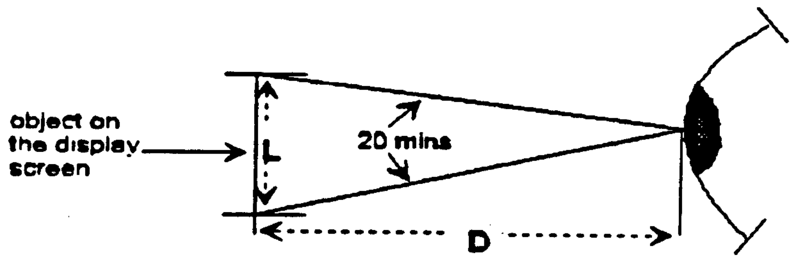


Figure 4-6. Visual Arc Subtended

4.3.4.2 Color-Coded Symbol Brightness

Ensure that color-coded symbols have a minimum luminance of one footLambert.

4.3.4.3 Refresh Rates

The minimum refresh rate for color-coded symbols should ensure a flicker-free display. Flicker-free display testing is described in American National Standards Institute (ANSI)/HFS Standard No. 100 (1988).

4.3.5 Map Graphics And Color

4.3.5.1 Functional Versus Decorative Color Coding

On map graphic displays, use color coding that provides a specific meaning to the user, rather than colors that are decorative only. These specific meanings should be used in accordance with appropriate standards. For example, the U.S. Army standard uses green for vegetation, brown for topographic relief, etc. Standards include U.S. Army FM 21-26, *Map Reading and Land Navigation* (1987); DIA *"DIA Standard Military Graphics Symbols Manual"* (DIAM 65-x) (Draft 1990); and *North Atlantic Treaty Organization (NATO) Standardization Agreement 2019, Military Symbols for Land Based Systems* (1990), available through the U.S. Navy.

4.3.5.2 Differences in Color Perceived Distance

The designer should be aware of how different colors focus at different distances relative to the user's retina as a result of wavelength. To the user, some colors will appear to be closer than others, especially the more saturated colors.

This page intentionally left blank.

REFERENCES

Paragraph	Reference
4.1	DoD (1992a)
4.2	Galitz (1994)
4.2.1.1	Williams (1987b) Appendix A p. A-1; Smith and Mosier (1986) para 4.0-6; Brown et al. (1983) p. 1-11; Shneiderman (1987) p. 327; Smith and Mosier (1986) para 2.5-1; Lickteig (1989) p.10; Brown et al. (1983) p. 1-1 & 1-11; Tullis (1988) pp. 393 & 336; Hamel and Clark (1986) p. 26; Slominski and Young (1988) p. 2
4.2.1.2	Brown et al. (1983) p. 1-1
4.2.1.3a	Smith and Mosier (1986) para 2.5-2
4.2.1.3b	Hamel and Clark (1986) p. 28; Slominski and Young (1988) p. 3-4
4.2.1.3c	Galitz (1984) p. 103
4.2.1.3d	Galitz (1994) p. 59
4.2.1.4a	Nes (1986) p. 105
4.2.1.4b	Smith and Mosier (1986) para 2.5-10; Lickteig (1989) p. 10; Brown et al. (1983) p.1-5 & 1-12; Shneiderman (1987) p. 327
4.2.1.4c	Bowser (1991) p. 16; Smith and Mosier (1986) para 2.5-11; Galitz (1984) p. 102
4.2.1.4d	Tullis (1988) p. 382
4.2.1.4e.	Galitz (1984) p. 102; Lickteig (1989) p. 9
4.2.1.4f	Brown et al. (1983) p. 1-4
4.2.1.5	Smith and Mosier (1986) paras 2.5-16 and 2.5-17
4.2.1.6	Brown et al. (1983) p. 1-11
4.2.1.7	Galitz (1984) p. 102
4.2.1.8	Shneiderman (1992) p. 80
4.2.1.9	Galitz (1994); Shneiderman (1992)
4.2.2	Galitz (1994); Shneiderman (1992)
4.2.2.1	Slominski and Young (1988) p. 2
4.2.2.2	Lickteig (1989) p.9; Brown et al. (1983) p. 1-10; Galitz (1984) p. 99 & 102; Tullis (1988) p. 382

REFERENCES (cont'd)

Paragraph	References
4.2.2.3	Smith and Mosier (1986) para 2.5-7
4.2.2.4	Galitz (1984) p. 103; Shneiderman (1987) p. 327
4.2.2.5	Smith and Mosier (1986) paras 2.5-13 and 2.5-15; Tullis (1988) p. 387; Shneiderman (1987) p. 336
4.2.2.6	Smith and Mosier (1986) para 2.5-14
4.2.2.9	Shneiderman (1992) p. 84
4.2.3.1a	Bowser (1991) p. 16; Tullis (1988) p. 385
4.2.3.1b	Shneiderman (1987) p. 327
4.2.3.2a	Williams et al. (1987b) Appendix A p. A-1
4.2.3.2b	Smith and Mosier (1986) para 2.5-3
4.2.3.2c	Nes (1986) p. 103; Shneiderman (1984) p. 104; Brown et al. (1983) p. 1-6
4.2.3.2d	Nes (1986) p. 101; Brown et al. (1983) p. 1-13; Shneiderman (1987) p.105; Tullis (1988) p. 398-399; Grabinger and Amedeo (1988) p. 198
4.2.3.2e	Shneiderman (1987) p. 327
4.2.3.2f	Tullis (1988) p. 395
4.2.3.2g	Galitz (1984) p. 102
4.2.3.2h	Shneiderman (1987) p. 327; Brown (1989) p. 28-29
4.2.3.3a	DoD (1985) p. 3-3
4.2.3.3b	Tullis (1988) p. 399; Galitz (1984) p. 104; DoD (1985) p. 3-3
4.2.3.4a	Shneiderman (1987) p. 104; Nes (1986) p. 112; Tullis (1988) p. 397; Brown et al. (1983) p. 1-9
4.2.3.4b	Galitz (1984) p. 184
4.2.3.4c	Galitz (1984) p. 184
4.2.3.4d	Shneiderman (1987) p. 184
4.2.3.4e	Bowser (1991) p. 16
4.2.3.4f	Bowser (1991) p. 16

REFERENCES (cont'd)

Paragraph	References
4.2.3.5a	DoD (1985) p. 3-2
4.2.3.5b	DoD (1985) p. 3-3
4.2.3.6	Smith and Mosier (1986) para 2.5-18
4.2.3.7	Smith and Mosier (1986) para 2.5-4
4.2.3.8	Smith and Mosier (1986) para 2.5-5; Galitz (1984) p. 103; Shneiderman (1987) p. 327
4.2.3.9	Smith and Mosier (1986) para 2.5-6; Shneiderman (1987) p. 327; Brown et al. (1983) p. 1-5
4.2.3.10	HFS ANSI 100 (1988)
4.2.3.11	HFS ANSI 100 (1988)
4.2.3.12	Williams (1987b) Appendix A p. A-2; MacGregor and Lee (1988) p. 10; Galitz (1984) p. 103; Shneiderman (1987) p. 336
4.3	Thorell and Smith (1990)
4.3.1	Lickteig (1989) p.10; Nes (1986) paras 4.2.3 and 4.2.4; Galitz (1984) p. 122; Brown et al. (1983) para 7.2; Lickteig (1989) p. 10; Shneiderman (1987) p. 341; Bailey (1982) p. 421; Lewis and Fallesen (1989) p. 20; Rosch (1994)
4.3.1.2a	Smith and Mosier (1986) para 2.6-29
4.3.1.2b	Shneiderman (1987) p. 72 and 337; Tullis (1988) p. 390
4.3.1.2c	Lewis and Fallesen (1989) p. 23; Nes (1986) para 4.1.1; DoD (1989) para 9-1.3.3; Galitz (1984) p. 126 and 127; Slominski p. 4; Matthews (1987); Sidorsky p. 6.3-15q
4.3.1.2d	Galitz (1994), pp. 377-402; Galitz (1984) p. 126-127
4.3.1.2d1	Sidorsky p. 2.3.6 q-6,7; Galitz (1984) p. 121; Bailey (1982) p. 43; Lewis and Fallesen (1989) p.25
4.3.1.2d2	Smith and Mosier (1986) para 2.6-30; Brown para 7.4; DoD (1989a) para 5.4.1.4.5.5; Bailey (1982) p. 63; Lewis and Fallesen (1989) p. 20; Lickteig (1989) p. 10
4.3.1.2d3	Brown et al.(1983) para 7.7.6
4.3.1.2d4	Bowser (1991) p.18; Lewis and Fallesen (1989) p. 21; HFS (1988)

REFERENCES (cont'd)

Paragraph	Reference
4.3.1.2d6	Bailey (1993)
4.3.2.1a	Smith and Mosier (1986) para 2.6-27
4.3.2.1b	Galitz (1984) p. 125; Lewis and Fallesen (1989) p. 22
4.3.2.1c	Galitz (1984) p. 123
4.3.2.1d	Lewis p. 22
4.3.2.1e	Smith and Mosier (1986) para 2.6-26, 31; Sidorsky p. 6.3-15 o-1; Brown para 7.6.1; Nes (1986) para 4.2.2; Galitz (1984) p. 122; DoD (1989) para 5.15.3.3.7; Lickteig (1989) p. 10; Shneiderman (1987) p. 339
4.3.2.2a	Bowser (1991) p. 18; Brown (1983) para 7.6.2; Galitz (1984) p. 125; Shneiderman (1987) p. 340; Lewis and Fallesen (1989) p. 21; Bailey (1982) p. 263
4.3.2.2b	Bailey (1982) p. 246
4.3.2.2c	Smith and Mosier (1986) para 2.6-32; Sidorsky p. 6.3-15 o-2 and 2.3.6 q-3, 4; Brown (1983) para 7.7.1; Galitz (1984) p. 125; DoD (1989) para 5.2.2.1.18; Shneiderman (1987) p. 340; Hamel p. 5; Bailey (1982) p. 246; Lickteig (1989) p. 10; U.S. Department of the Army (1985b)
4.3.2.3a	Smith and Mosier (1986) para 2.6-28; Brown (1989) para 7.1; Galitz (1984) p. 127; Lickteig (1989) p. 10; Smith and Mosier (1986) para 2.6-28; Chao (1987) p. 361; Lewis and Fallesen (1989) p. 20
4.3.2.3b	Galitz (1984) p. 127; Nes (1986) para 4.2.2 and 4.2.5;Slominski p. 4; Shneiderman (1987) p. 338
4.3.2.3c	Sidorsky p. 6.3-15 o-3; Galitz (1984) p. 125; Lickteig (1989) p. 10; Bailey (1982) p. 421; Shneiderman (1987) p. 71
4.3.2.4a	Lewis and Fallesen (1989) p. 22
4.3.2.4b	Galitz (1984) p. 126
4.3.2.4c	Galitz (1984) p. 126
4.3.2.4d	Shneiderman (1987) p. 341; Snyder (1988) p. 465; Matthews (1987) p. 23

REFERENCES (cont'd)

Paragraph	Reference
4.3.2.4e	IBM (1984) p. 19
4.3.2.5a	Galitz (1984) p. 127
4.3.2.5b	Galitz (1984) p. 127
4.3.2.5c	Galitz (1984) p. 127
4.3.2.6	Sidorsky p. 2.3.6 q-3 and 4; DoD (1989) p. 256; U.S. Department of the Army (1985) p. 2.2
4.3.2.7	Lewis and Fallesen (1989) p. 23
4.3.2.7a	Snyder (988) p. 465
4.3.2.7b	Smith and Mosier (1986) para 2.6-34; Brown (1983) para 7.7.5; Galitz (1984) p. 127
4.3.2.8	Sidorsky p. 6.3-15 o-4; Brown para (1983) 7.6.1; Nes (1986) para 4.2.4; Galitz (1984) p. 123; Lickteig (1989) p. 10; U.S. Department of the Army (1985) p. 2-2
4.3.2.9	Lewis and Fallesen (1989) p. 22
4.3.2.10	Lewis and Fallesen (1989) p. 22; Shneiderman (1987) p. 341; Snyder (1988) p. 465; Sidorsky para 2.3.6; Thorrell p. 2 and 3
4.3.3.1	Smith and Mosier (1986) para 2.6-25
4.3.3.2	Smith and Mosier (1986) para 2.4.8- 7
4.3.4.1	Durrett (1987) p. 186; Van Cott and Kinkade (1984) p. 47
4.3.4.2	Breen et al. (1987) p. 207
4.3.4.3	Breen et al. (1987) p. 209; HFS ANSI 100 (1988)
4.3.5.1	Olson (1987) p. 207; U.S. Department of the Army (1987); DIA 1990; NATO 1990
4.3.5.2 Olson (1987)	p. 20

This page intentionally left blank.

5.0 WINDOWS

A window provides the visual means by which the user can interact with an application program. A window displays the results of the command or data input by keyboard, mouse, or other device. A window display screen is analogous to a window in a wall that allows one to see into a room; the window display screen allows the user to see into a software program. A window is typically rectangular and can cover part or all of a display screen. In addition, multiple windows on a display can be open at one time. Figure 5-1 illustrates an OSF/Motif window. Windows for other GUI, such as OS/2 and Windows, have similar though not identical characteristics. Arrows and labels identify key parts of the window. Section 5.0 provides general guidelines for windows. Refer to commercial GUI style guides for specific window design details and explanations of attributes and terms used to describe the actions, warnings, and information presented to the user.

Commercial GUI designs provide a number of basic functions, allowing the user to control window operations. While each GUI provides its own specific functions, the following are typical examples. By opening and closing a window, a task or application is started, stopped, or removed from the screen. Scrolling allows the user to view the information within a window, including that which is outside the normal boundaries of the window. Windows can be stacked on top of each other like paper on a desk. Good designs should provide the user the capability to access available GUI functions when they are required.

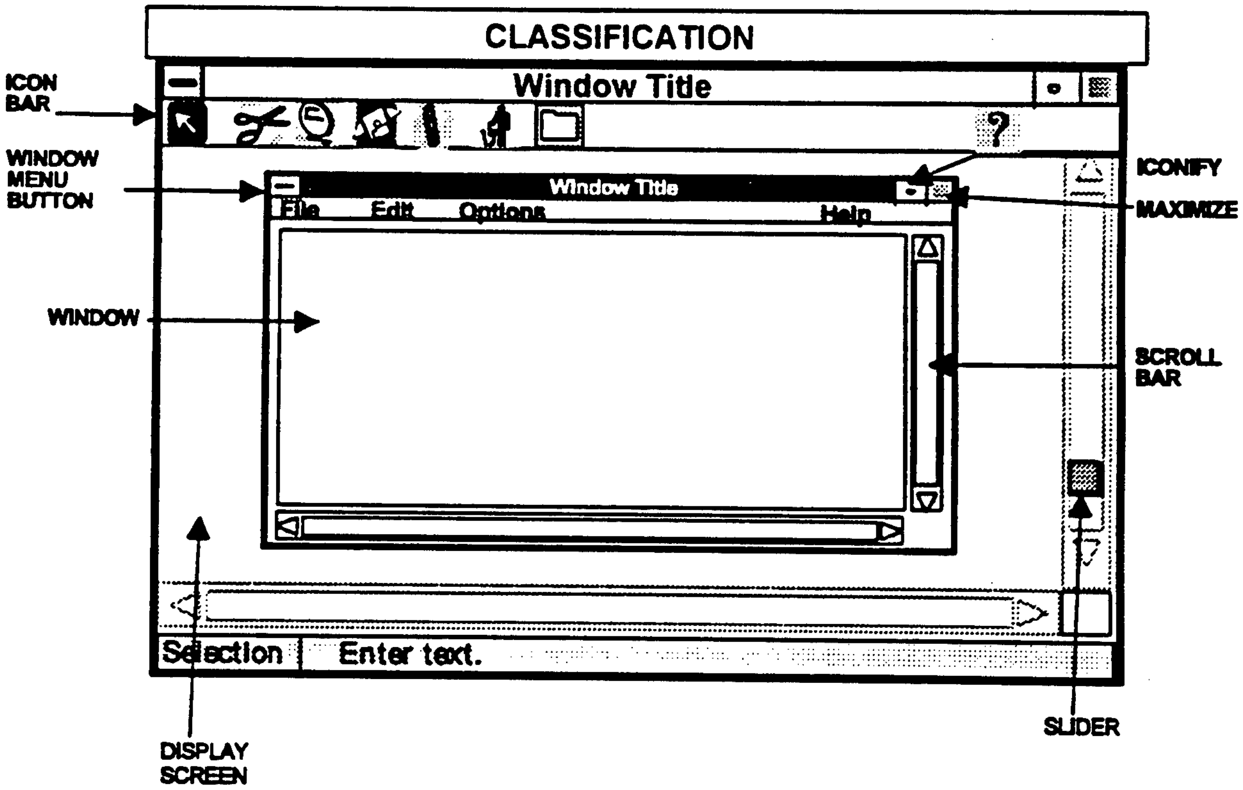


Figure 5-1. Example of a Typical Windowing Screen

The two basic approaches to simultaneous window presentation are tiling and overlapping. In the tiling approach, multiple windows do not overlap but lie on the same plane. Their borders are flush, and developers usually limit the primary control operations to designation and scrolling, while limiting or blocking basic functions, such as opening and closing or moving and sizing. Figure 5-2 illustrates the tiling type of window design.

Using the overlapping method, windows are presented on multiple planes and appear to be 3-dimensional. Windows can overlap or even obscure each other, like pieces of paper on a desk top. The window made active by the user will appear on top, pushing inactive windows to the back. The user normally has access to all previously discussed control functions to control overlapping windows. Figure 5-3 illustrates this type of window design.

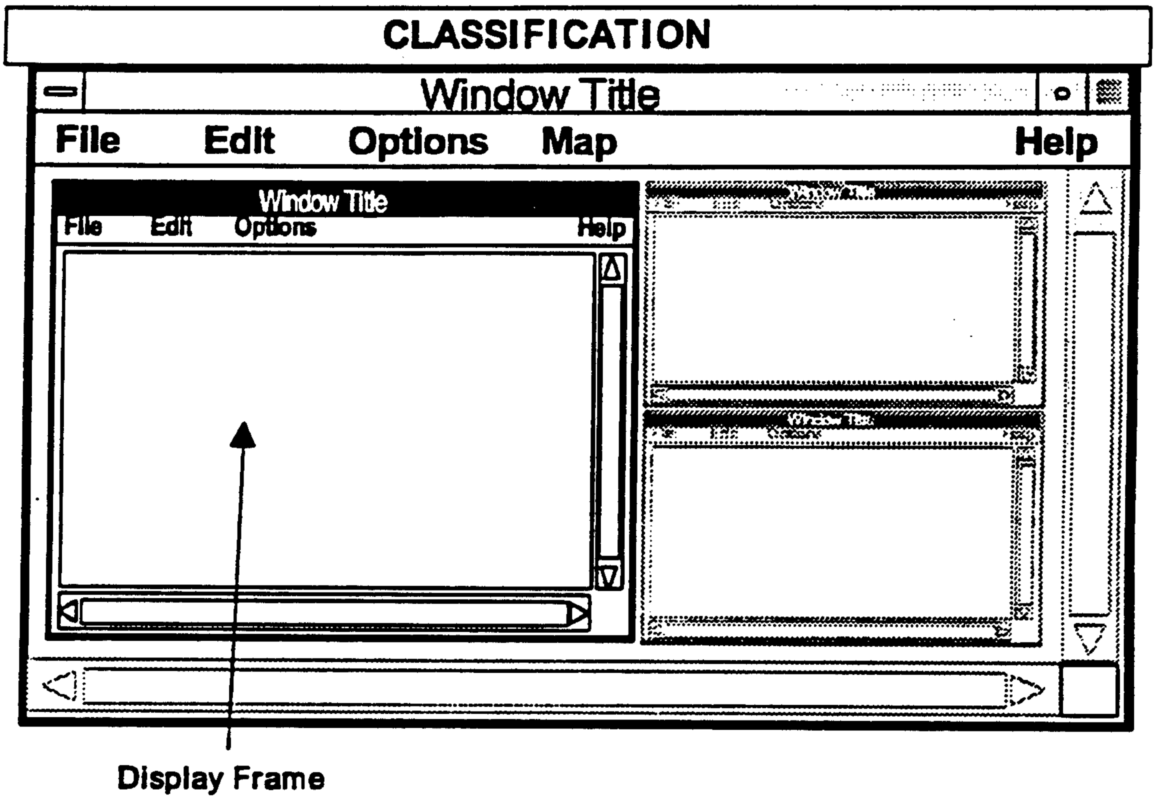


Figure 5-2. Example of the Tiling Approach

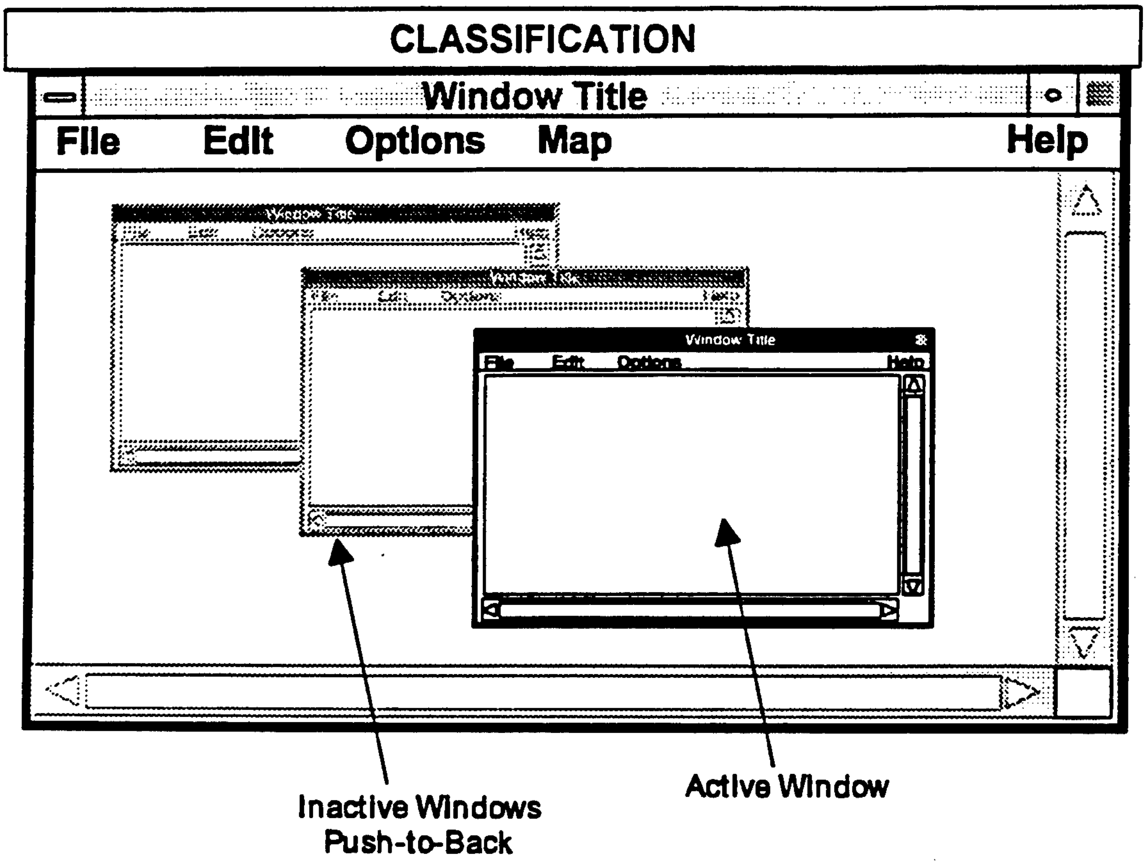


Figure 5-3. Example of the Overlapping Approach

The design of application windowing interfaces should begin with the screen design principles given in Section 4.0 of the *Style Guide* and should address the basic window attributes using the appropriate commercial style guide. The depth and breadth of research on the impact of windows on user performance are not as great as they are for other design areas (Billingsley 1988).

In general, the following generic guidelines should be applied to window GUIs.

- Be consistent in how the windows look and "act."
- Recognize the limitations that the specific system hardware imposes on the usefulness of windowing software. For example, ensure that the display device has the resolution and size to properly support a windows approach to information presentation. When the hardware will not adequately support the windowing environment, alternate hardware should be considered.

- Ensure that the central processor unit (CPU) has the power, in terms of memory and speed, to effectively use a windows approach. Without a proper CPU, slow system-response time will significantly degrade the speed of information presentation. Again, when the hardware will not adequately support the windowing environment, alternate hardware should be considered.

- The windows interface is especially important when the user needs to perform multiple tasks or see different sets of data concurrently.

- Each open window requires system resources in terms of memory and processing speed. Through experimentation, determine a limit on the maximum number of windows that can be effectively opened for each system.

5.1 WINDOW BASICS

A clarification of window and screen-related terms is provided in Figure 5-4. The Glossary, Appendix B, contains additional term definitions. The *Style Guide* should not conflict with commercial GUI style guides, and the actual definition to be used by the developer should comply with the selected commercial GUI.

5.1.1 Basic Window Appearance

The displayed window appearance is determined by the selected GUI and related commercial style guide. Specific domains further define window appearance, such as intelligence, where the basic CMW window components are added to the commercial window design. The classification bar displayed as the top line of the basic window and the optional input information label displayed at the bottom of the screen are additional features supported by the intelligence community CMW operating system rather than by the GUI style selected. However, from the CMW application designer's viewpoint, classification bar and input information label are displayed in the same manner as other window controls (e.g., the title bar). Appendix A includes a detailed description of the fields that make up these security bars. Until the CMW becomes an integral component of the DoD system architecture, each DoD organization should adhere to its own security standards.

5.1.1.1 Title Bar

The appearance of the title bar and associated controls are determined by the selected GUI. The creation of the titles within a title bar is subject to general positive design principles. If the application contains multiple primary windows (e.g., to display different files), then the window title should include the application name and the name of the currently displayed file.

Category	Term	Definition	Reference
Screen-related terms	computer screen	Hardware monitor total display area	Figure 8.5
	display frame	Area of a screen identified for design, that includes more than one window or object	Figure 5.2
	display screen	Area of the hardware screen used for display	Figure 5.1
	dialog box	Screen display box containing a message requesting additional information from the user	Figure 5.7
	input focus	Applies to window that actually receives user input. Input focus may be explicit or implicit (see glossary).	Figure 5.11
	window	Typically rectangular display that provides a visual means for interaction with an application	Figure 5.1
Window-related terms	multiwindow	Simultaneous display of several windows on the computer screen	Figure 5.2
	push-to-back	Process of moving a window to the background	Figure 5.3
	overlapping	Windowing system in which one window covers a portion of another	Figure 5.3
	tiling	Windowing approach in which multiple windows do not overlap, rather, all lie on the same plane	Figure 5.2
Window/screen parts	cursor	Visual mechanism to mark, on-screen, where current input or output is to happen	Figure 9.3
	pointer	Graphic on the screen display that represents the mouse or trackball position	Figure 5.11
	resize border	Window border that, if selected, allows user to resize the window	Figure 5.14
	scroll bar	Rectangular bar that may be along the right edge or bottom of a window Clicking or dragging in the scroll bar causes the view of the document to change.	Figure 5.1
	slider	Part of the scroll bar that indicates what part of the file contained in a window is being viewed	Figure 5.1
Menu-related terms	menu	List of options available within a software application	Figure 6.5
	icon bar	Horizontal or vertical layout of icons used as buttons to quickly access frequently used commands and macros	Figure 5.1
	menu bar	Horizontal menu, usually at the top of the screen, that contains menu titles	Figure 5.8
	menu button	A button in a standardized location used for window management functions (i.e., close, move, resize)	Figure 5.1
	hierarchical menu	Method of organizing menus in layers. The secondary or tertiary menus are stored within a primary menu.	Figure 6.7
	pop-up menu	Lists of options that appear on the display screen in the form of a window	Figure 6.3
	pull-down menu	Lists of options attached to a selection on a menu bar	Figure 6.2

Figure 5-4. Window and Screen-Related Terms

Some general considerations that apply to creating titles are the following:

- Include the name of the application in the title, followed by a colon, followed by the name of the currently displayed file (e.g., Editor: Myfile.txt)
- Center the title
- Distinguish the title by a visual attribute (e.g., boldface type)
- Use application and function name to identify an open window, not system-level window name (e.g., messages:e-mail as opposed to ATCCS:e-mail)
- If the selected commercial GUI allows, enable the window title to display the version number of the application, but do not use the window title area to display any messages.

5.1.1.2 The Window Menu Button

The window menu button for Windows and Motif systems is located in the upper left-hand corner of the title bar (see Figure 5-1). This button provides a standard location for window management functions (e.g., close, move, and window resizing functions). A more detailed explanation of the functions and features supported by the window menu button can be found in the relevant GUI style guides. The principle of using a consistent location and shape for standard controls should be applied to all applications.

5.1.1.3 Reducing the Window to an Icon

In standard GUI styles, users can iconify windows. For example, if the user is not actively using a base window but wishes to maintain easy access to it, or if the window is active but does not require user interaction for extended periods, these windows can be iconified. If a window is reduced to an icon, the window is removed from the screen, and the application controlling the window is represented as an icon. Application processing can then continue in the background, as if the window were still displayed on the screen. This capability to iconify is available to application developers as part of any standard GUI implementation and should be used as appropriate to good user interface design.

5.1.1.4 Expanding a Window to its Full Size

Expanding a window to its full size (maximizing) increases the size of the window to the maximum specified by the application. The maximize methods used by the various GUIs include selecting a maximize button, selecting a maximize function from the window menu button, or depressing the maximize accelerator keys with the window focus appropriately selected. Windows can also be expanded to full size by dragging (see Paragraph 5.1.2) the resize borders or resize corners. The capability to easily expand a window to full size is available to application developers as part of any standard GUI implementation and should be used when appropriate to good user interface design.

5.1.2 Dragging the Window

Dragging refers to a user's ability to reposition windows or window borders on the screen. Dragging a window moves it to a different position on the computer screen. As the window is dragged (or moved), a "ghost" outline of the window should move with the pointer. The window should move to the position of the outline when the procedure is complete (e.g., mouse button is released).

5.1.3 Scroll Bars

The scroll bar is a special type of control that makes it easy for the user to view or page through objects such as documents, drawings, and spreadsheets too long or wide to be displayed in the application area, also called a pane. Scroll bars also aid users in panning graphic map displays in the north/south and east/west directions. Scroll bars give users the capability to navigate through documents without paging one window at a time. This interface capability is available to application developers as part of any standard GUI implementation and should be used when appropriate to good user interface design for all windowing applications.

5.1.4 Application Area

The application area or pane is the part of the window where applications display and collect data and where users perform most application tasks. For example, if a user is working with a text editor, the application area could contain the document to be edited. When using a windowing interface, the application area should be clearly and consistently identified to the user.

5.1.5 Message Area

The message area (or footer) is reserved for noncritical application messages that should not suspend processing. Use the left side of the message area for short-term messages, such as "Incorrect format - field requires numeric data. Please reenter." Use the right side of the message area for medium-term messages, such as "Page 4 of 29."

5.1.6 Resizing The Window

The application suggests the initial size of its window to the window manager. Because work and preferences vary, users should generally be able to alter the size of windows. Resizing windows is performed through "hooking" the edge or corner and dragging the cursor to reduce or increase the window size, or by using buttons typically located in the upper right corner of a window. Resizing a window normally increases or decreases the size of the window frame, not the scale of the data within the window. For example, if a window containing a text document is enlarged, more lines of data may be seen, but the text itself does not enlarge.

The specific resizing behavior of a window is determined by the commercial GUI selected. However, the following are general principles to use when resizing windows:

- In the minimum height of a window, allow enough room for at least the classification bar, title bar, and menu bar (control area).
- Design the application logically to accommodate the resizing function. Include important information in the upper left-hand corner of the window.
- When a user resizes a window, ensure that only the size of the window's borders changes, not the size of graphics, text font size, relative position of the data, or the controls within the borders. The normal result will be an increase in the amount of viewable text or number of objects in the window. An exception might occur in imagery manipulation where the user may require the image to rescale (magnify) with the window frame.
- Ensure that resizable windows are easily distinguishable from those that cannot be resized, such as the system window.

5.1.7 Window Controls

Controls and their labels represent application functions in windows and dialog boxes. See Subsection 6.6 on Dialog Boxes/Pop-Up Windows.

- Controls should mimic the physical items they represent (e.g., switches or buttons) by providing feedback before, during, and after their selection by a user. For example, a button that the user has chosen should appear to be pushed in.
- Window controls are usually activated using the SELECT button on the pointing device. However, users who interact with the application using only the keyboard should have equivalent functionality. Ensure that control appropriate to the selected commercial GUI is available. Control examples include the "TAB" or arrow keys, allowing the user to move between controls and using the Return/Enter key to invoke the default of the indicated control. Also, when mnemonics are available to application developers as part of a standard GUI implementation, the keyboard user should be provided with mnemonics for each control.
- Graphic display of a control should use shape, shading, outline, and (when appropriate) color to aid the user in identifying the active control area. When the control background is the same as the area surrounding the control, care must be taken to clearly mark the control area for easy identification.

5.1.8 Window Colors/Patterns/Audio Signals

The proper use of color, background patterns, and sound may significantly aid the user. This section provides recommendations for using these features.

- Ensure that color is always redundant with some other visual attribute; color should not be provided as the only means of visual distinction.

- On both color and monochrome displays, use background patterns to highlight, group, or clarify relationships and to add extra meaning.

- For quick and accurate interpretation, use colors sparingly and ensure that these colors match user expectations (see Subsection 4.3).

- Ensure that colors that may be changed are not "hard coded" into applications. When appropriate, users should have the option to select their own color schemes (see Subsections 4.3 and 14.1).

- Some colors have strongly associated meanings that the designer must be aware of and use, not abuse. For example, a user may assume that a red control button has critical or irreversible consequences. Thus, avoid red for noncritical buttons, as it may inhibit the user from exploring them. Some common color meanings are as follows:

–	Red	Stop, alarms, errors, danger, critical consequences
–	Yellow	Warning, caution, approaching critical
–	Green	Normal, safe, within normal range, proceed
–	Blue	Cold, water, noncritical items
–	Gray	Inactive, unavailable options or actions.

- Use both color and sound for messages that require user acknowledgment. Display critical messages using red (i.e., borders, text background) and continue the audio alarm until the user responds. Display noncritical messages (e.g., "Printer error. Please check printer and retry or cancel") using yellow (i.e., text background, graphic, border, etc.) accompanied by a short audio alert.

- Do not use spectral extremes (e.g., red and green, see Subsection 4.3) on a display at the same time, close together. Colors at considerably different wavelengths appear to vibrate when placed together.

- When data are color-coded, provide a legend (e.g., "Orange = Required Field") at the bottom of the window. Limit color codes to four per window and no more than seven per application.

- Use the same color scheme (i.e.,window background, foreground, etc.) for all windows of an application. Repeated use of the same color for similar user interface components or data types allows quick association of elements.

- White text on a black background produces halation, or the spreading of light, making the text less readable. Displaying text in multiple colors also makes text less readable and should be used only if the addition of colors provides significant additional meaning.

- Ensure that the computer screen and window pane (workspace) background is appropriate to the expected lighting conditions (see Subsection 4.3) and that it is a neutral color.

- Ensure that the application window borders contrast sufficiently to stand out from the screen background. At the same time, provide a neutral back-ground for the application data to ensure readability. Muted pastels are recommended.

- In general, the larger the object, the less saturated or deep its color should be to avoid eye fatigue.

- CMW Classification Bar colors are listed below. Environments that use DoD security classification colors should restrict background colors that match the domain-level definition of these display colors.

–	Green	Unclassified
–	Blue	Confidential
–	Red	Secret
–	Orange	Top Secret
–	Yellow	Sensitive Compartmented Information.

5.2 WINDOW DESIGN

5.2.1 General Guidance

5.2.1.1 Hardware Limitations on the Use of Windowing

When the interface is affected by limitations of the selected hardware platform(s), the developer needs to design the windowing interface accordingly. Hardware limitations to consider include:

- Small screen size, resulting in frequent manipulation of the screen by the user

- Slow processing speed, resulting in slow operation of real-time applications performed by the computer

- Low screen resolution, resulting in less effective visual coding, especially for graphical interface presentations such as symbols and icons.

5.2.1.2 Flexibility of Window Specification

A key to the effective design of a windowing user-computer interface is the flexibility the user has in customizing window content and format. A balance must be achieved between user-specified windows and preformatted windows.

- When the need to view several different types of data jointly cannot be determined in advance, allow a user to specify and select separate data windows that will share a single display frame.

- Where the information required for decision-making may vary according to the situation, allow the user to specify what information to include in a display.

- When content of particular operational displays can be determined during interface design, provide the user with preformatted windows, such as standard message texts for data entry and display.

- Allow the user to display several of these windows concurrently, according to the operational need.

5.2.1.3 Temporary Window Objects

Temporary window objects (e.g., pop-up menus or data, option menus, data filters) are especially effective for providing a menu of alternatives for field entry in preformatted tactical messages and database queries.

- When it is necessary to temporarily add requested data or other features to a current display, provide window objects for that purpose.

- Ensure that a temporary window object does not completely cover the active window, thereby obscuring critical control information and command entry widgets, soft keys, or other activation points.

- When a window object temporarily obscures other displayed data, ensure that obscured data are not permanently erased but will reappear when the object is later removed.

5.2.1.4 Number of Allowable Open Windows

To ensure that system response time is not compromised, design into the interface a defined upper limit on the number of windows allowed to be open at one time.

5.2.1.5 Window Physical Design

- Avoid visual clutter in designing windowing systems.

- For tiled window systems, minimize the clutter at the edges caused by scroll bars, etc. Figure 5-5 illustrates a cluttered window design for tiled windows. Figure 5-2 illustrated an uncluttered display.

- For overlapping window systems with multiple windows, keep back-ground pattern neutral, rather than use complex patterns. Figure 5-6 illustrates a cluttered display; Figure 5-3 illustrated an uncluttered display.

- When a display window must be used for scanning data exceeding more than one line, ensure that the window can display more than one line of data.

- When the system provides an area within a window for command entry, messages, or prompts, place this area as specified in the commercial style guide, or if not specified in commercial style guide, place this area at the bottom of the window display.

- Dialog boxes should be designed to comply with the selected commercial GUI to ensure that they look and function consistently for all applications and systems. To achieve this, follow these recommendations. See example in Figure 5-7.

  - Control buttons used to input a command from a dialog box should be located consistently, for example at the bottom of the window. If the selected commercial GUI allows this placement, this is consistent with the user's natural task flow.

  - The button used to input the selected or default command (usually an OK) should be located consistently, normally on the left side of the box. The CANCEL button is usually located on the right side. Any additional control buttons should generally be located between the OK and CANCEL buttons.

  - The individual commercial GUI style guide is the primary source for the specific placement of controls in a dialog box.

5.2.2 Window Control

Control refers to how the user manipulates the window, not how the application operates within the window. Guidelines for the design of window control fall into five basic topics: general guidelines, opening and closing, moving, sizing, and scrolling.

5.2.2.1 General

- When a user may perform application control actions (such as command entry) while working within a window, ensure that those control actions will be consistent from one window to another.

- Ensure the means provided to the user for controlling (after initial display) the size, location, and characteristics of superimposed window objects operate consistently from one display to another for each type of object.

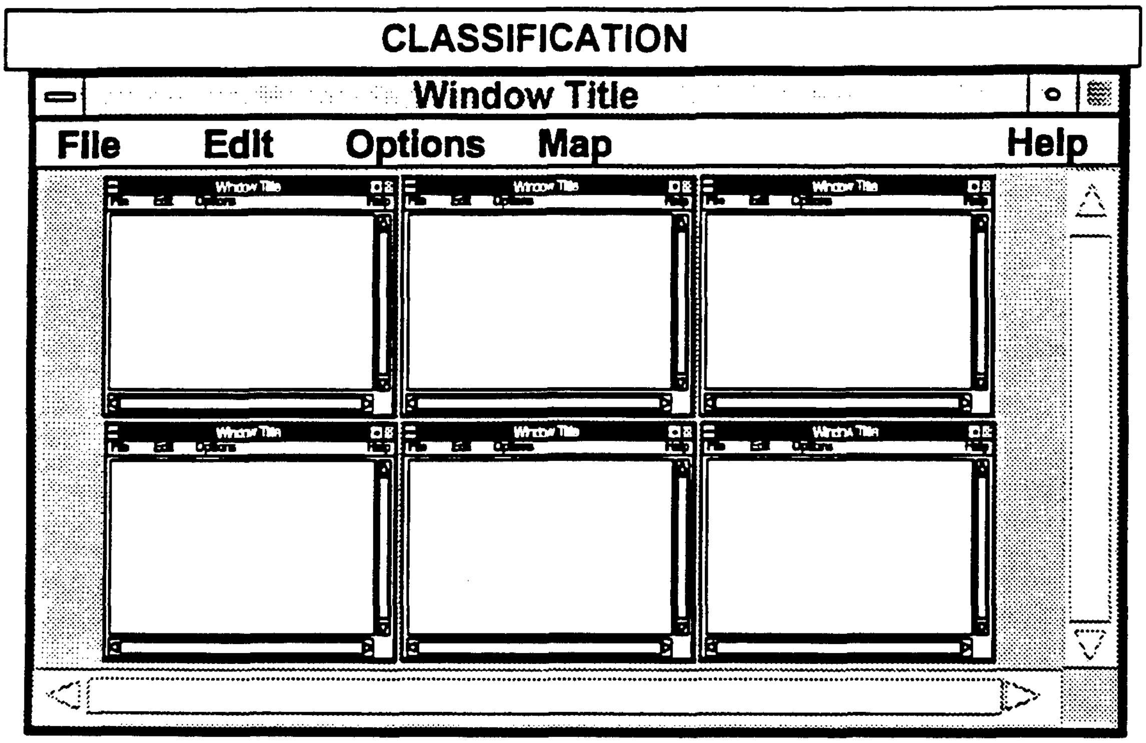


Figure 5-5. Example of Cluttered Window Design for Tiled Windows

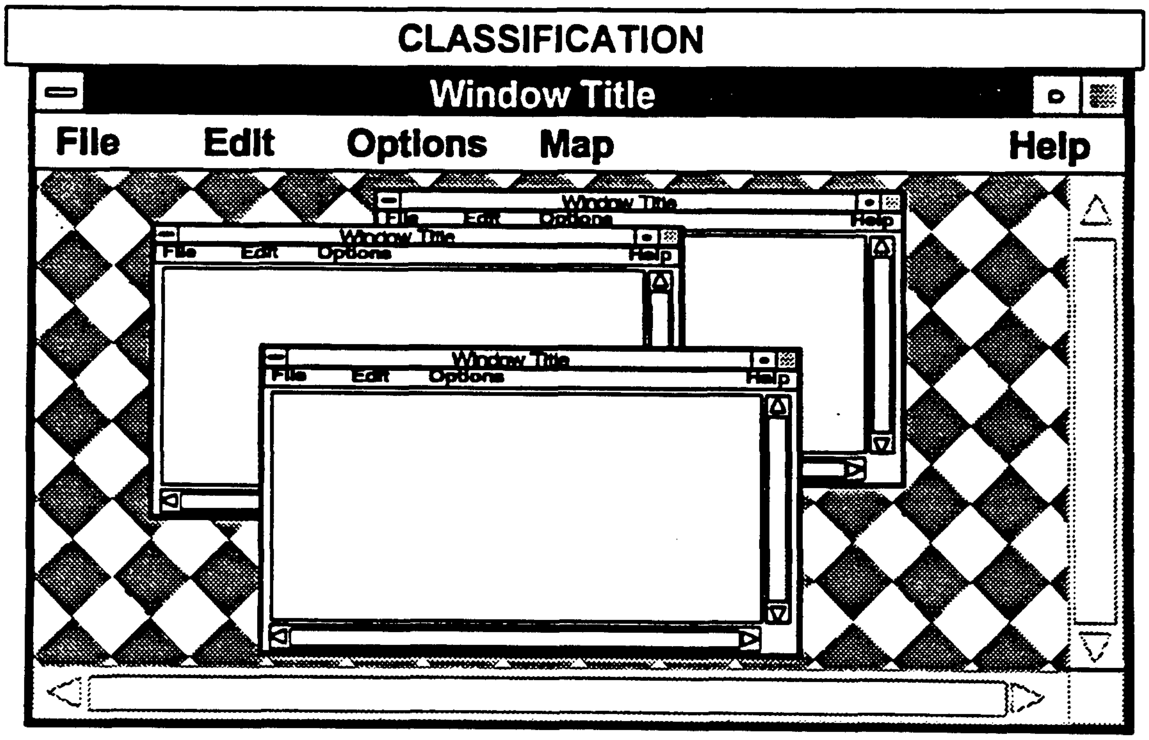


Figure 5-6. Example of Cluttered Window Design for Overlapping Windows Induced by a Complex Background Pattern

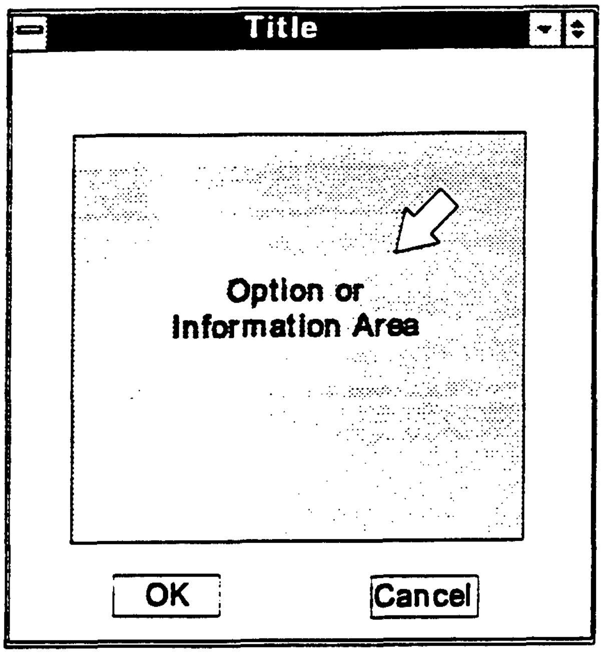


Figure 5-7. Example of a Dialog Box Design

- Provide an easy means, such as iconization or closing, for the user to suppress the display of window objects.

- Provide a separate menu bar for each application window, where different applications are operating concurrently in open windows (e.g., multitasking). See the example in Figure 5-8.

5.2.2.2 Opening and Closing Windows

Windows can be opened or closed by menu selections, or a close-button widget (i.e., a small, push-button control object usually located in the upper left corner of a window), or opened from an icon or minimized to an icon. When designing the opening and closing operations, consider the following guidelines:

- The software should provide an animated depiction of opening and closing a window by portraying the window shrinking to an icon and vice versa. This helps the user relate the window, icon, and action (see Figure 5-9).

- When a main applications window is closed by the user, all associated subordinate windows and dialog boxes should also close.

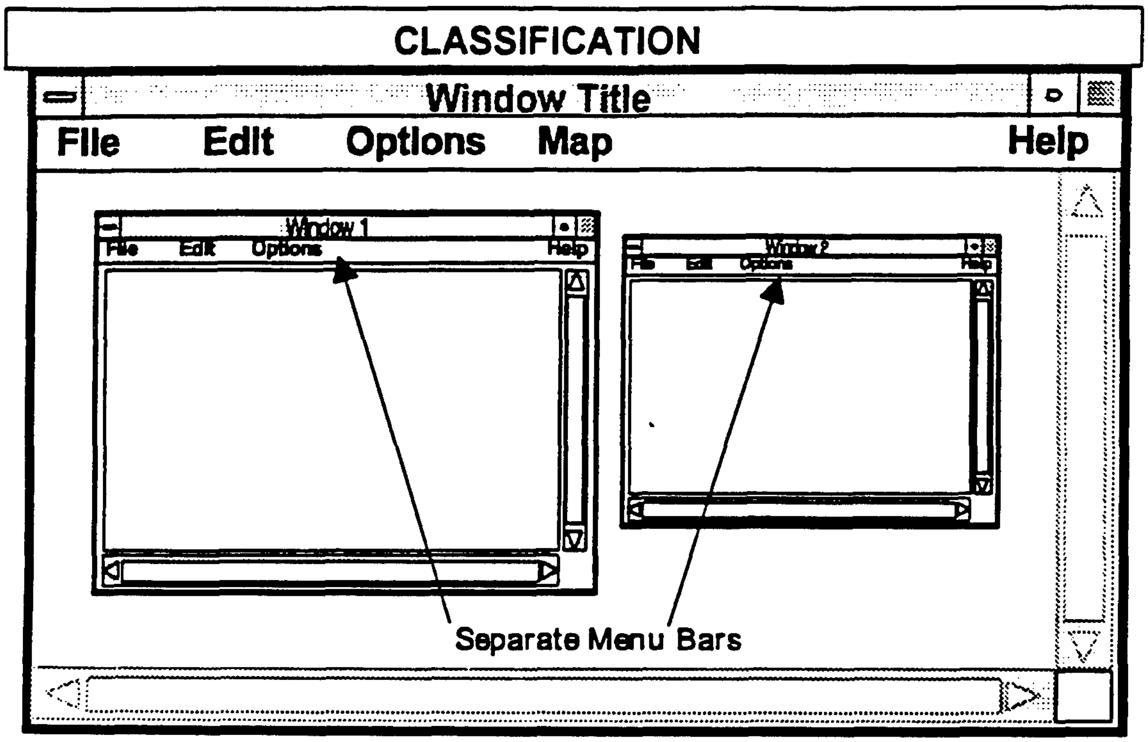


Figure 5-8. Example of Different Applications with Separate Menu Bars

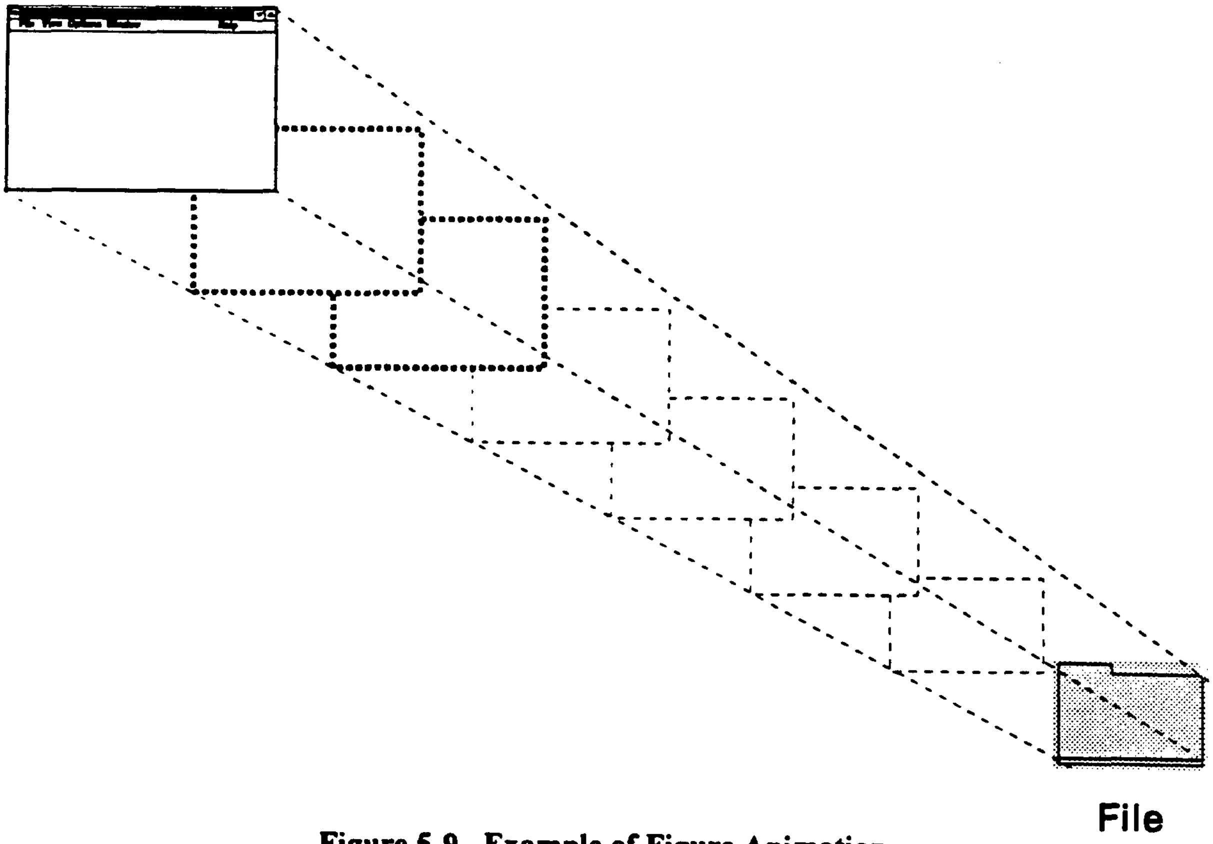


Figure 5-9. Example of Figure Animation

5.2.2.3 Moving Windows

- Provide either full movement of the window (see Figure 5-10) or move an outline, leaving the window visible on the screen.

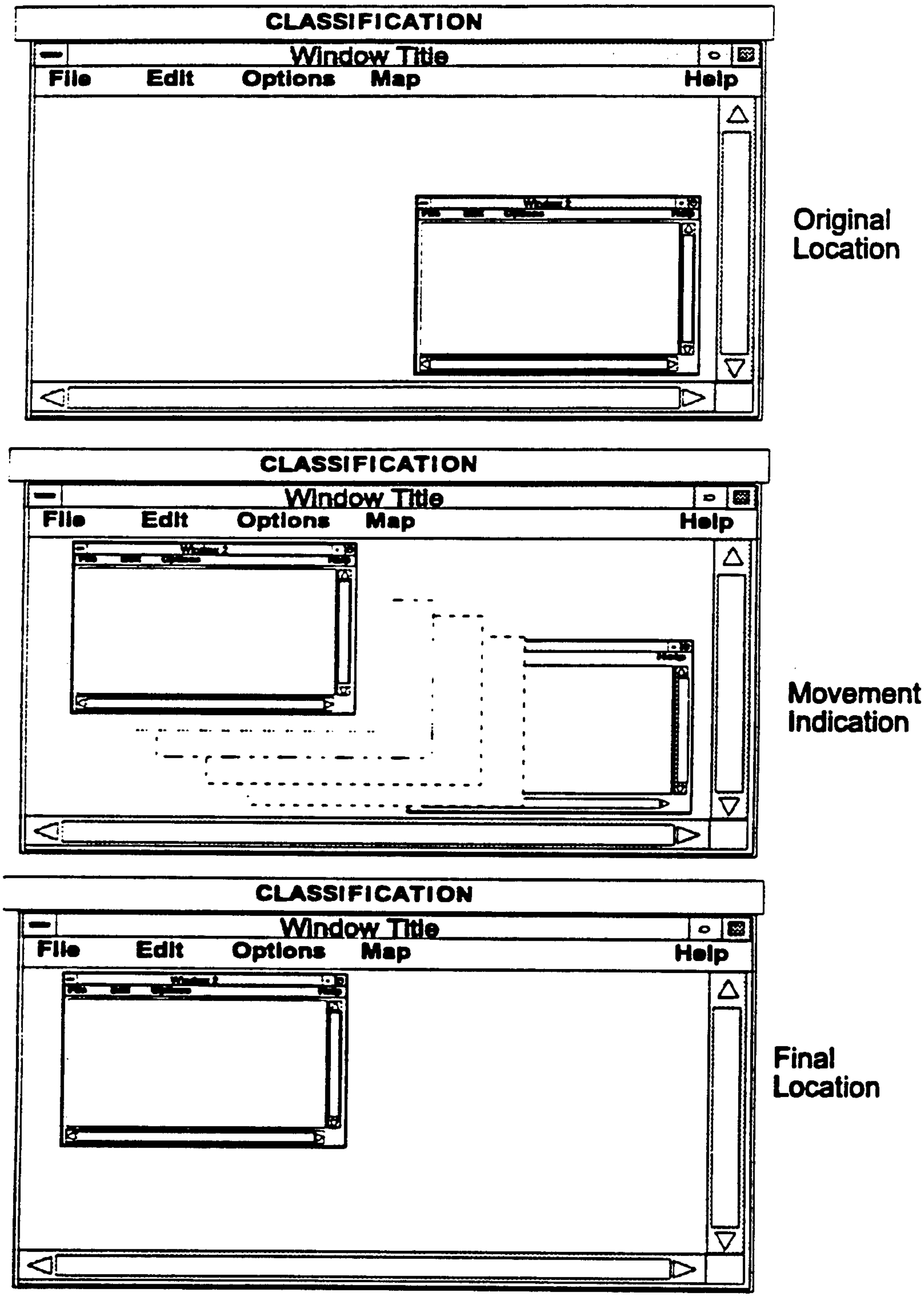


Figure 5-10. Example of a Screen Move

- When the user must select a specific move function to relocate a window on a screen, ensure that the cursor indicates this by a change in shape. Figure 5-11 illustrates one type of cursor change.

5.2.2.4 Resizing Windows

- Provide system protection against obscuring critical control information during window manipulation, especially during user maximization of the window. This means system protection for both the data being retrieved through dialog boxes and system-level control information, such as alert indications. See Figures 5-12 and 5-13.

- When a window is resized, ensure the window contents remain visible during the resizing to provide a visual indication of the effect on the window contents (e.g., visibility and integrity of the image), rather than providing just an outline. Keeping the contents visible will reduce the number of steps required by the user (e.g., resize, view, etc.).

- Resizing of tiled windows by a user is not recommended. If resizing is absolutely required for a tiled window system, ensure that the system automatically resizes all other open windows when one is resized by the user.

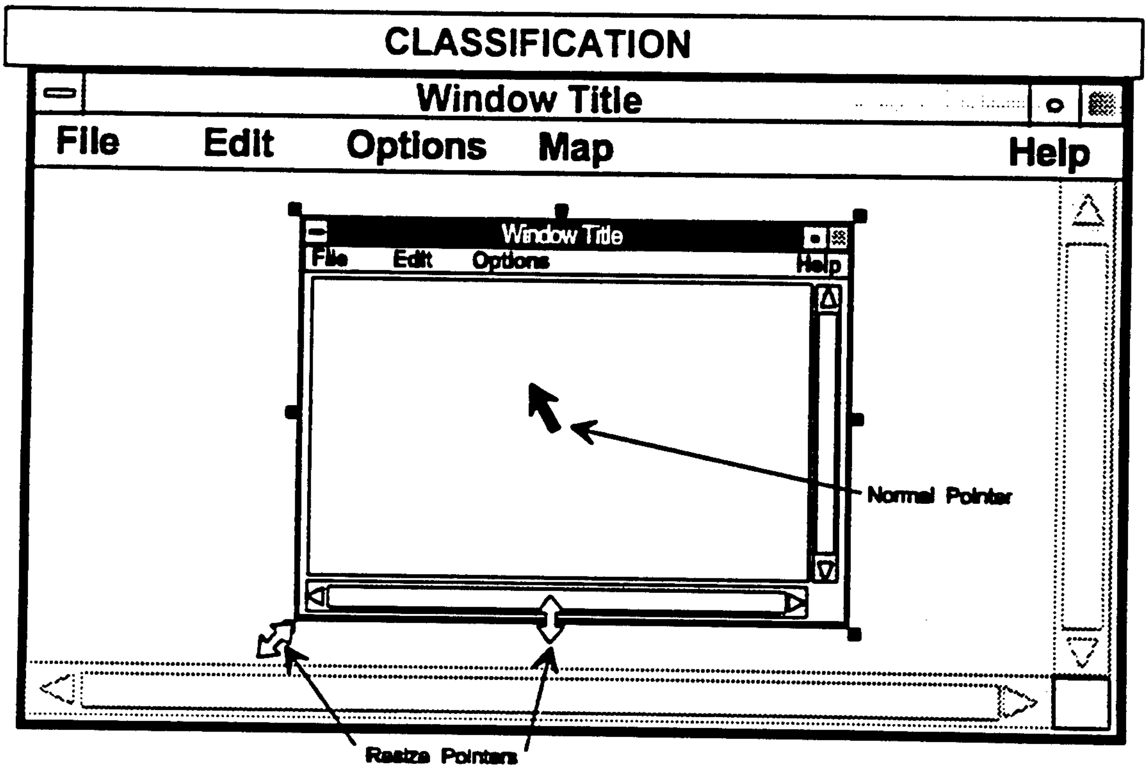


Figure 5-11. Example of a Pointer Changing Shape

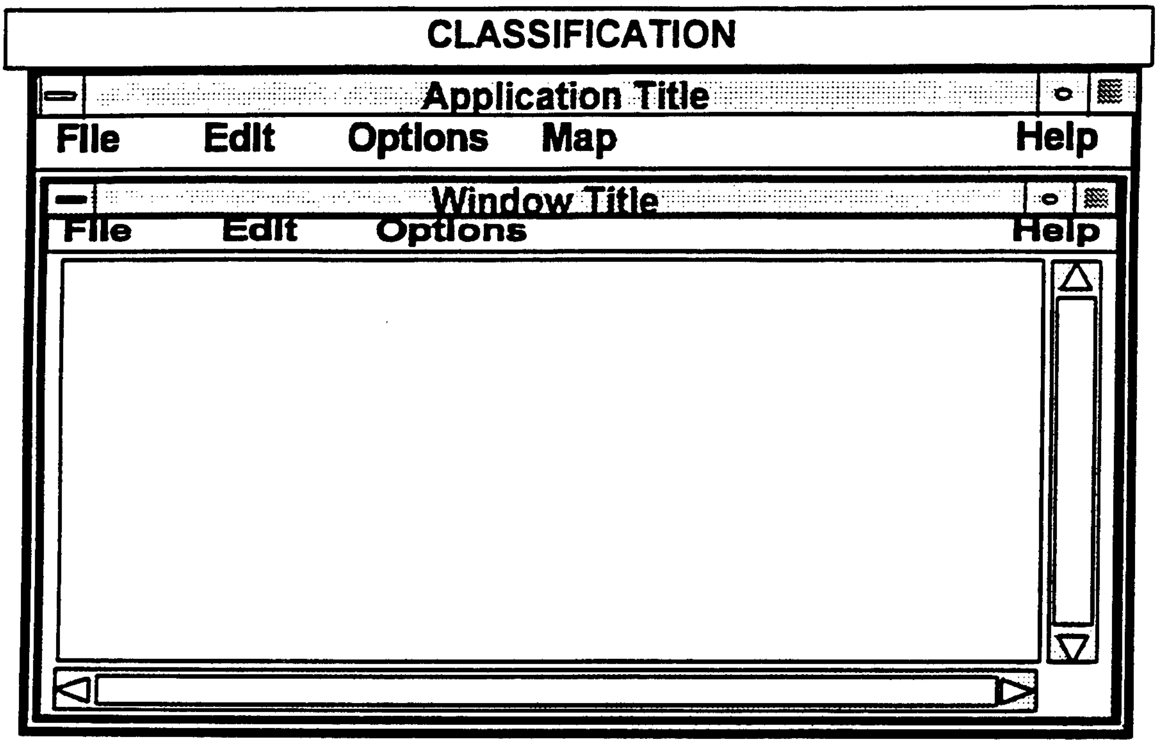


Figure 5-12. Maximum Size of Window

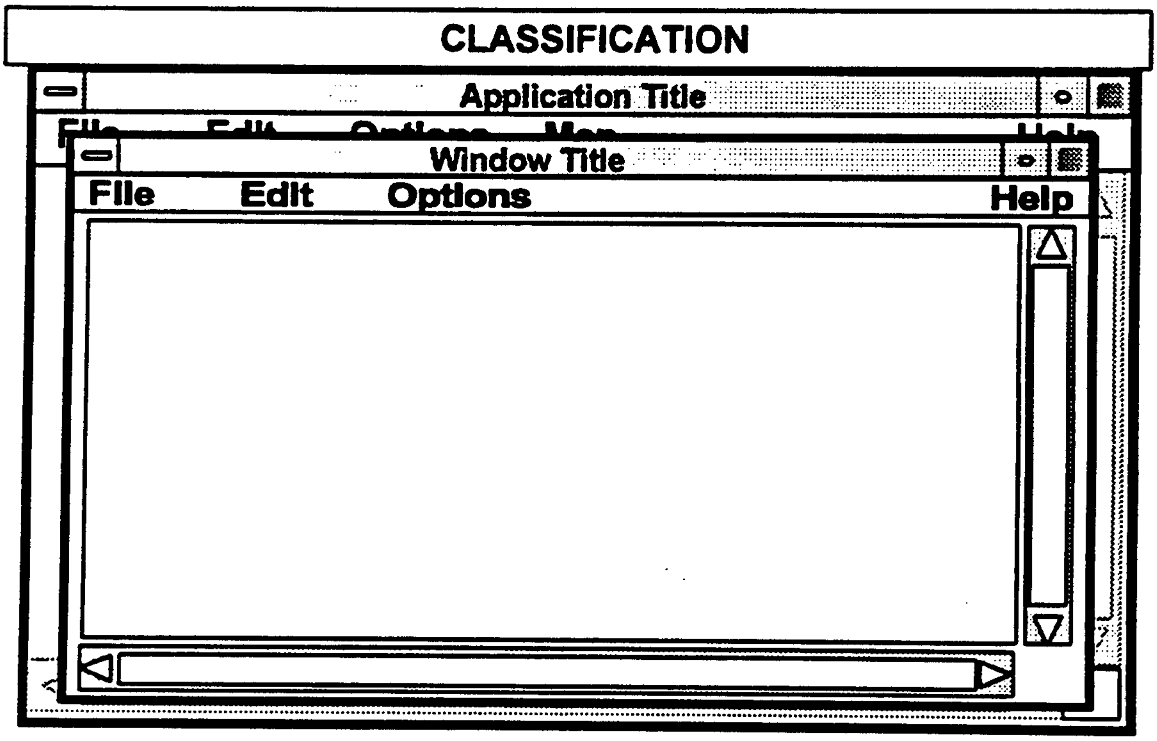
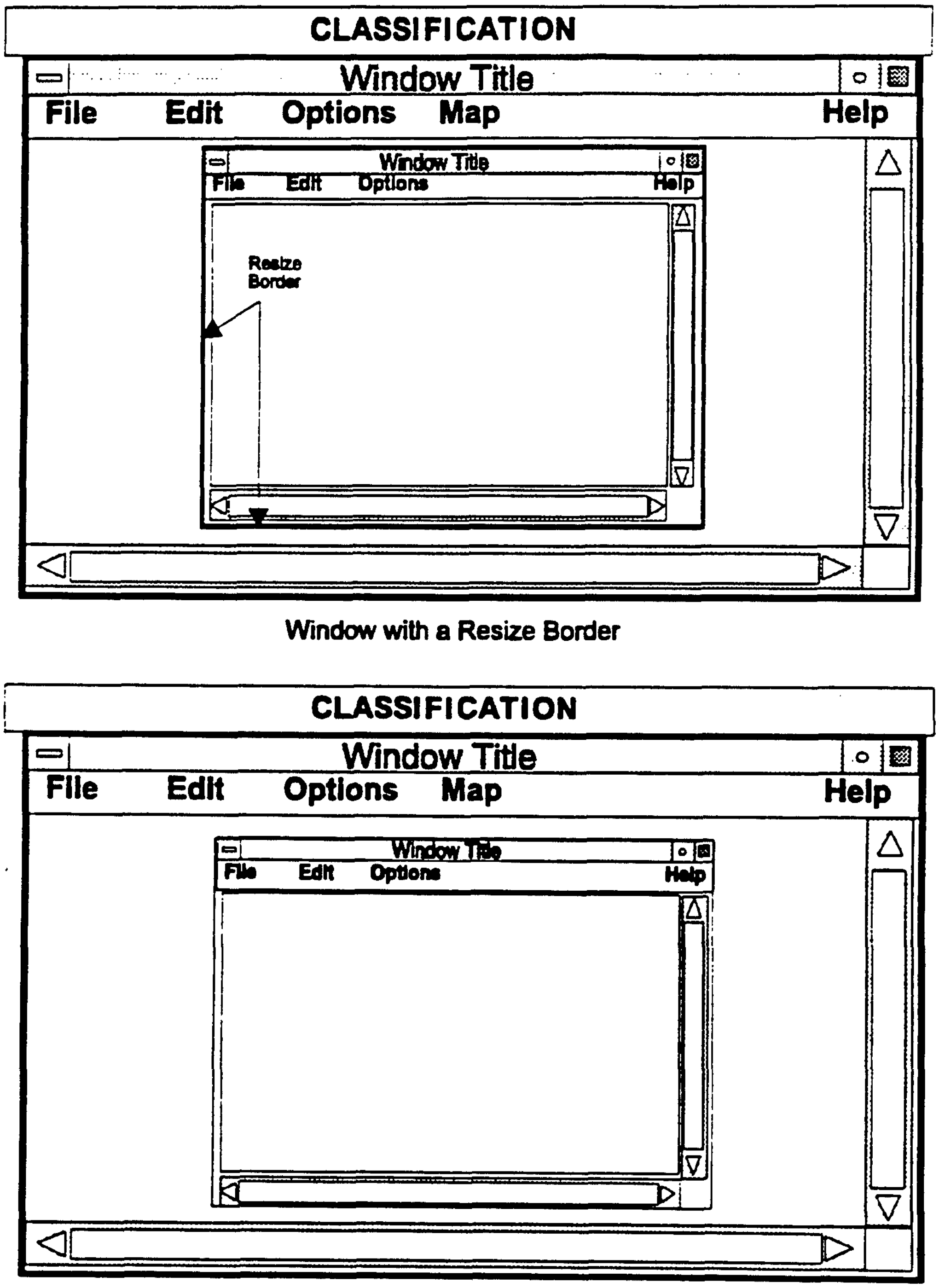


Figure 5-13. Window Size Too Large, Covering Critical Information

- Most windows have a resize border (see Figure 5-14) located at the peripheral edge. If a window cannot be resized, the resize border should be removed to provide a positive indication to the user that the window size is static.



Window with a Resize Border

Window with Resize Border Removed

Figure 5-14. Resize Border Removal

5.2.2.5 Scrolling Windows

Scrolling a window can be performed two ways: 1) move the window over the data, where upward movement of the scroll bar causes data to appear to move down; 2) move the data past the window, where upward movement of the scroll bar causes data to appear to move up.

- For scrolling, the system should move the window over the data, as this is consistent with the general convention in industry, such as found in the *OSF/Motif Style Guide*.
- The distance the slider moves on a scroll bar should be proportional to the distance traveled through the file in a window to assist the user in determining current location relative to the total file.
- Design window displays to preclude excessive scrolling. If possible, use a single screen for the full display, unless it causes reading difficulty due to reduction of screen character size.
- Do not display the scroll bar if scrolling is not necessary.

5.2.3 Designation

Designation is the process of selecting and indicating with visual cues which window the user can use. This window is called the input focus window.

5.2.3.1 Positive Indication of the Active Window

When more than one window is open, provide the user with a clear, positive indication of the active window by means of a more complex border, subtle change in color hue, or labeling change. This active window should be distinct yet not distract the user's attention from window activity (see Figure 5-15).

5.2.3.2 Easy Shifting Among Windows

If several window objects are displayed at once, provide some easy means for the user to shift among them to select which window will be currently active. For example, shift the cursor with the mouse, then press the mouse button to designate the active window.

5.2.4 Labeling

5.2.4.1 Labeling Windows

Assign an identifying label to window objects, dialog boxes, or subordinate windows. This label should briefly describe the contents, purpose of the window, or the menu path (e.g., Messages:e-mail:outbox).

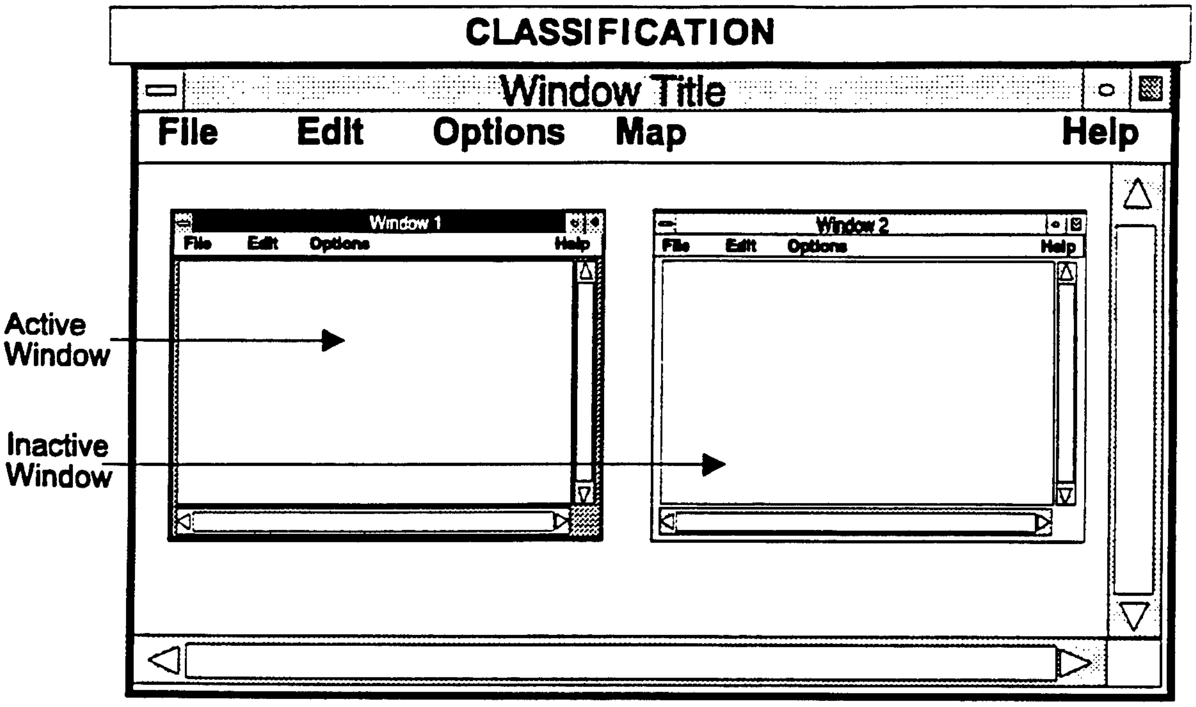


Figure 5-15. Example of Active Window Designation

5.2.4.2 Format of Subordinate Window Labels

Ensure that titles of subordinate windows match menu selection items from the supraordinate window menu.

5.2.4.3 Window Titles

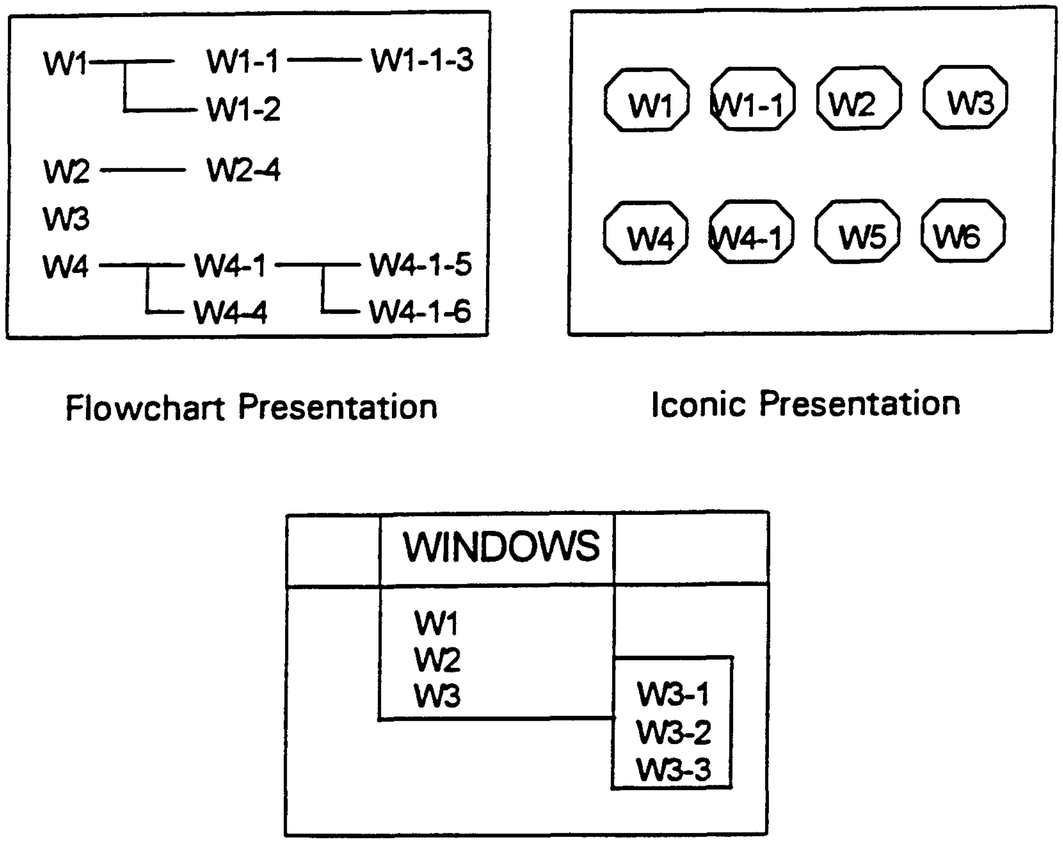
Locate window titles consistently.

5.2.5 Open Window Navigation

Navigation, in terms of windows, refers to the user's ability to move among the various windows that are open on a display.

5.2.5.1 Open Window Map

Applications should, when using an overlapping window structure, provide a user-requested iconic or text map/indication of all open windows to allow the user to easily identify all open (especially the hidden) windows. Figure 5-16 shows three presentations of an open window map.



Flowchart Presentation

Iconic Presentation

Pull-Down Window Presentation

Figure 5-16. Examples of Open Window Maps

5.2.5.2 Active Designation from Open Window Map

Provide the user the capability to designate the active window through the iconic or text open window map by highlighting the window representation.

5.2.5.3 Expanded Window Explanation of Open Window Map

If possible, allow the user to query an open window map for expanded information (e.g., date created, size, description of subject or application, etc.) on the file or application operating in the window. This information is usually accessed through HELP.

5.2.5.4 Window Forward Function With Window Map

When an iconic or text map is provided for determining the numbers and names of open windows in an overlapping system, allow the user to bring a window forward from the map without having to resize or move other windows.

REFERENCES

Paragraph	References
5.0	Avery and Bowser (1992); Billingsley (1988); DISA/CIM (1992)
5.1	DISA/CIM (1992)
5.2.1.1	Billingsley (1988) p. 416
5.2.1.2a	Avery et al. (1990); Smith and Mosier (1986) paras 2.5.2.5-3 and 2.5.8
5.2.1.2b	Avery et al. (1990); Smith and Mosier (1986) para 2.5.2.5-2
5.2.1.3a	Smith and Mosier (1986) para 2.5.2.5-1
5.2.1.3b	Slominski and Young (1988); Avery et al. (1990)
5.2.1.3c	Smith and Mosier (1986) para 2.5.2.5-10
5.2.1.4	Billingsley (1988) p. 421
5.2.1.5a	Avery et al. (1990); Billingsley (1988) p. 421; Bowser (1991) p 12; Smith and Mosier (1986) para 2.5-11; OSF (1990) Chapter 7
5.2.2.1a	Smith and Mosier (1986) para 2.5.2.5-9
5.2.2.1b	Smith and Mosier (1986) para 2.5.2.5-4
5.2.2.1c	Smith and Mosier (1986) para 2.5.2.5-5
5.2.2.1d	Nielson (1987) p. 247
5.2.2.2a	Billingsley (1988) p. 428
5.2.2.2b	OSF (1990) p. 3-4
5.2.2.3a	Billingsley (1988) p. 428
5.2.2.3b	Billingsley (1988) p. 428
5.2.2.4a	Slominski and Young (1988); Avery et al. (1990)
5.2.2.4b	Billingsley (1988) p. 429
5.2.2.4c	Billingsley (1988) p. 421; OSF (1990) p. 4-5; Avery et al. (1990)
5.2.2.4d	OSF (1990) p. 3-6
5.2.2.5a	Billingsley (1988) p. 430; OSF (1990)
5.2.2.5b	Billingsley (1988) p. 430
5.2.2.5c	Slominsky and Young (1988) p. 5; Avery et al. (1990)

REFERENCES (cont'd)

Paragraph	References
5.2.3.1	Billingsley (1988) p. 431; Lewis and Fallesen (1989) p. 94; Smith and Mosier (1986) paras 2.5.2.5-7
5.2.3.2	Smith and Mosier (1986) para 2.5.2.5-8
5.2.4.1	Smith and Mosier (1986) para 2.5.2.5-6 and 2.5-10; OSF (1990) p. 3-4
5.2.4.2	Slominski and Young (1988) p. 2, 54 p. 3-4
5.2.4.3	Slominski and Young (1988) p. 2
5.2.5.1	Billingsley (1988) p. 420
5.2.5.2	Billingsley (1988) p. 431
5.2.5.3	Billingsley (1988) p. 420
5.2.5.4	Billingsley (1988) p. 420

6.0 MENU DESIGN

Using menus as a dialog is widespread within computer systems. Menus are frequently used in conjunction with other interactive methods, such as direct manipulation.

Using menus as a dialog has advantages, a major one being that it requires little training or sophistication on the part of the user. A user needs know only the meaning of each menu option, then is guided step by step through the operation of the system. The number of keystrokes required to access a system function may also be reduced, thereby speeding the user-to-computer transaction.

On the other hand, using menus as a dialog has disadvantages. It does not enhance retention of commands and may actually increase response time for the more experienced user. Menus may take up a large part of the display surface. In addition, for complex sequences, using menus may require an extensive menu tree structure, and the user may easily become lost navigating through a complex menu tree.

A number of different types of menuing techniques are available to the designer, including pull-down, pop-up, and sequential display. Pull-down and pop-up menus tend to be used more in direct manipulation types of dialog. Sequential display, where a control action causes another menu to overwrite the previous menu, is used more in text-based systems. All these types of menuing techniques can be hierarchical, or branching, in nature.

The following pages provide detailed design guidelines for menus used in operational systems. To ensure a high level of user performance with menus, the designer should be aware of the following general guidelines:

- Consider choosing menus when:
  - tasks involve choosing among a constrained set of alternative actions
  - tasks require infrequent entry of data
  - the user may have little training
  - the computer response is relatively fast
  - tasks require infrequently used commands
  - command sets are so large that the user is not likely to commit all commands to memory.
- Design the menu tree structure broad and shallow, rather than narrow and deep. Keep the number of top-level options large, with a small number of sublevels.
- Consider the experienced user and provide a mechanism by which the menu structure can be bypassed using a direct command.

The designer should also note that some of the guidelines discussed in the following paragraphs may be more appropriate for designing sequential display menus than for menus used in direct manipulation. The designer should use judgment regarding which approach to take.

The designer should conform to a single interface style, such as "Motif," throughout an application. Varying interface styles confuses the user. Widgets, or graphical objects that are components of a user interface, or graphics as menu item selectors should be unique and clearly identifiable by the user.

6.1 GENERAL

6.1.1 Consider Response Time and Display Rate

If computer response time is long, create menus with a larger number of items. If display rate is slow, create menus with fewer items to reduce display time.

6.1.2 Instructions and Error Messages

Indent menu instructions and error messages and place them in the same position on the screen so the user knows where to look for this information.

6.1.3 Explicit Option Display

When entries for any particular computer transaction consist of a small set of options, show those options in a menu added to the working display, rather than require a user to remember them or access a separate display.

6.1.4 Stacking Menu Selections

For menu selection by code entry, when a series of selections can be anticipated before the menus are displayed, permit the user to combine those selections into a single stacked entry. Stacking refers to stringing multiple commands together and executing them with one action.

6.1.5 Menus Distinct From Other Displayed Information

If menu options included in a display are also intended for data review and/or data entry, ensure they are distinct from other displayed information. Locate menu options consistently; use consistent visual cues for their special function.

6.1.6 Menu Bars

Menu bars provide system functions in a bar across the top of the display screen. The following guidelines apply to menu bars.

6.1.6.1 Using Menu Bars

A menu bar is best used with standard sized screens (12-19 inches). With large-screen displays, the distance the pointer is required to travel may be too great to be effective. In this context, large-screen displays are defined as intended for multiple viewers, including projections and theater-type displays.

6.1.6.2 Visibility of Menu Bar Options

Ensure that menu bar options remain constantly visible (see Figure 6-1).

6.1.7 Pull-Down Menus

Pull-down menus, as illustrated in Figure 6-2, are lists of options attached to a selection on the menu bar that remain visible until the user takes action. Use pull-down menus instead of pop-up menus when pointer position on the screen is not important for information/option retrieval.

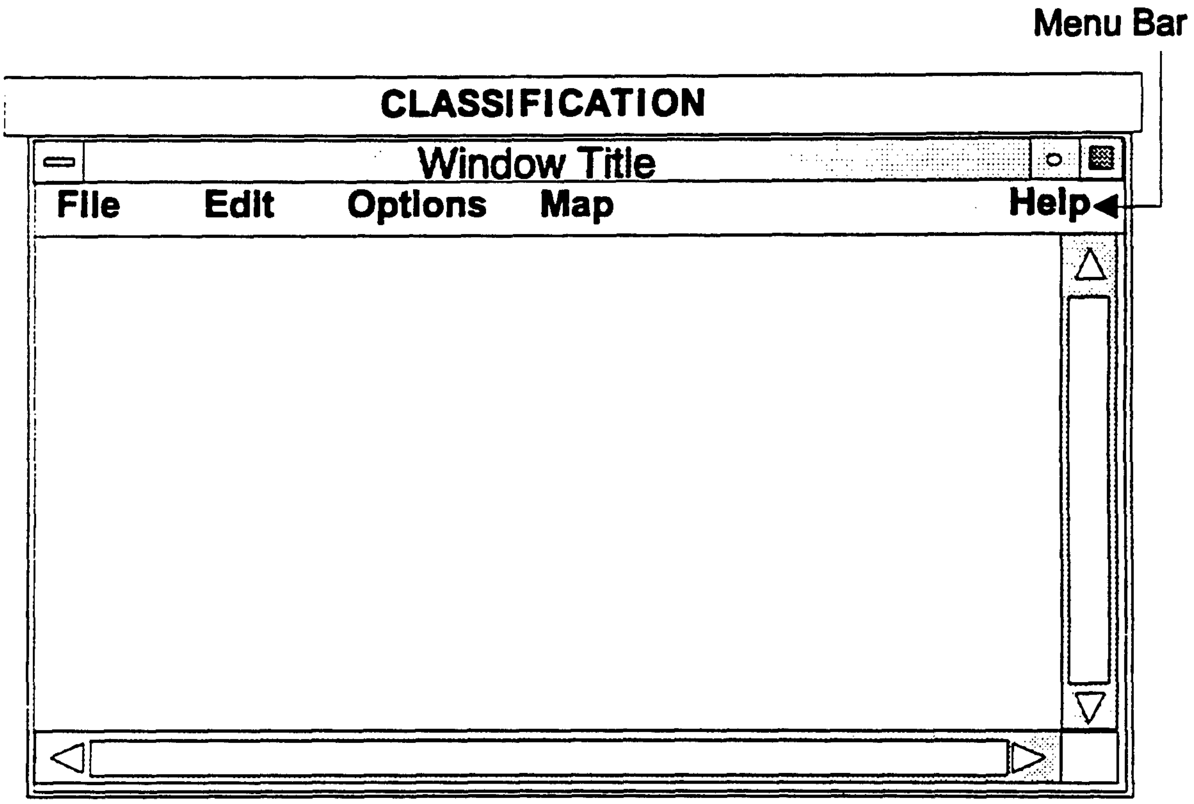


Figure 6-1. Example of Menu Bar

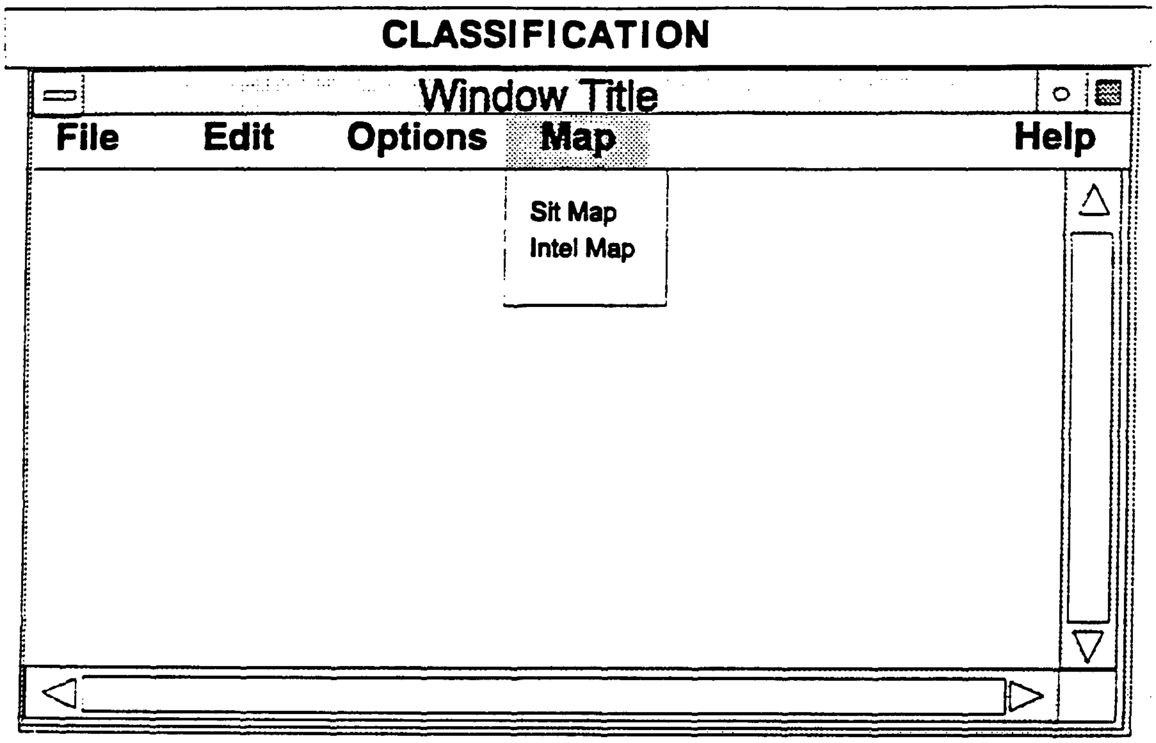


Figure 6-2. Example of a Pull-Down Menu

6.1.8 Pop-Up Menus

Pop-up menus are lists of options that appear on the display screen in the form of a window (see Section 5.0). Pop-up menus are specific to their area on the display; each window or object may have its own individual pop-up menu. The following guidelines should be used when designing pop-up menus.

6.1.8.1 Pop-Up Menu Location

Ensure that pop-up menus are connected to pointer location and pop up near the object or higher level menu being manipulated. See map overlay, Figure 6-3.

6.1.8.2 Selecting Options From Pop-Up Menus

Two methods to select from a pop-up menu are: 1) hold the button down while traversing options, then release to make the selection, or 2) move the pointer and press the button again for the selection. Use the second method when a choice is made to use only one selection method. Although it involves more keystrokes, it is less error-prone. It is acceptable to enable the application to allow either method and give the choice to the user.

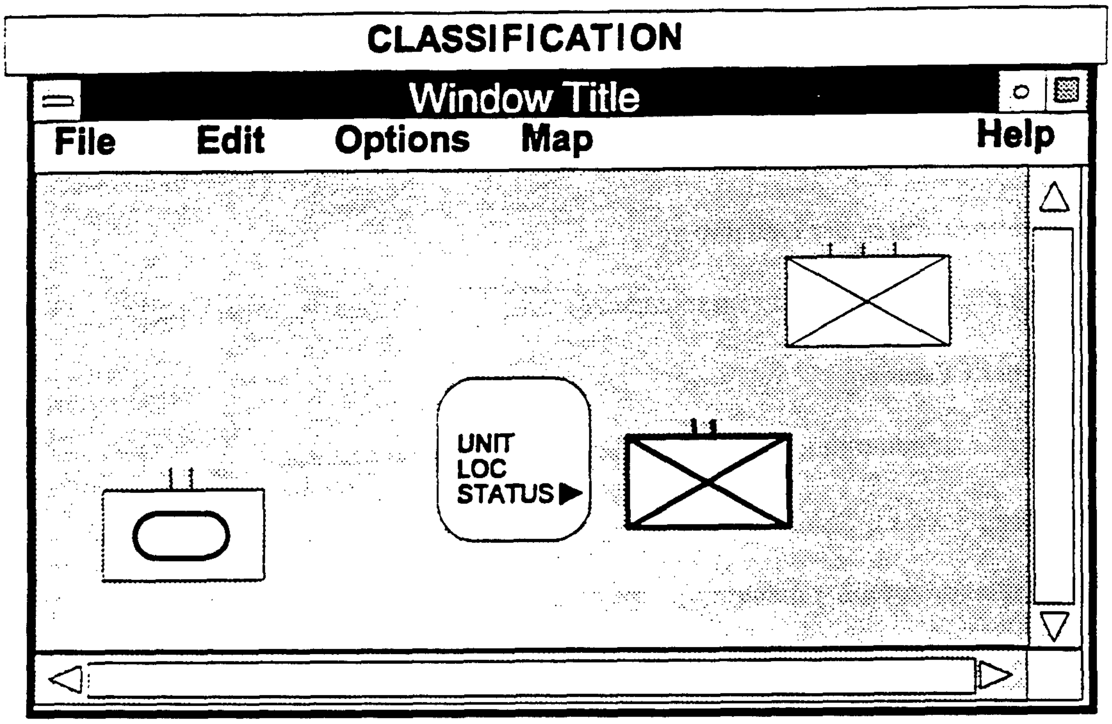


Figure 6-3. Example of a Pop-Up Menu

6.1.8.3 Selection Highlighting

When an option has been selected from a pop-up menu, ensure that it remains highlighted.

6.2 FORMAT

6.2.1 General

6.2.1.1 Menu Format

Keep lists of menu and submenu items brief (no more than five to nine options), arranged in separate columns, aligned, and left-justified.

6.2.1.2 Consistent Display of Menu Options

When menus are provided across different displays, design them so option lists are consistent in wording and order.

6.2.1.3 Logical Grouping of Menu Options

Format a menu to indicate logically related groups of options, rather than an undifferentiated string of alternatives.

6.2.1.4 Logical Ordering of Grouped Options

If menu options are grouped in logical subunits, display those groups in a logical order. If no logical structure is apparent, display the groups in the order of their expected frequency of use. See the example in Figure 6-4.

6.2.1.5 Sequence or Frequency Ordering

For a small number of menu items, use sequence or frequency to determine menu order.

6.2.1.6 Alphabetic Ordering

For a large number of menu options, use alphabetic ordering of menu items.

6.2.1.7 Numbering Menu Options

When task order is important, list menu options by number, not by letter.

6.2.1.8 Display of Options

In designing a menu for a GUI, display unavailable menu items in a visually distinct manner. Refer to Paragraph 8.3.3.5.

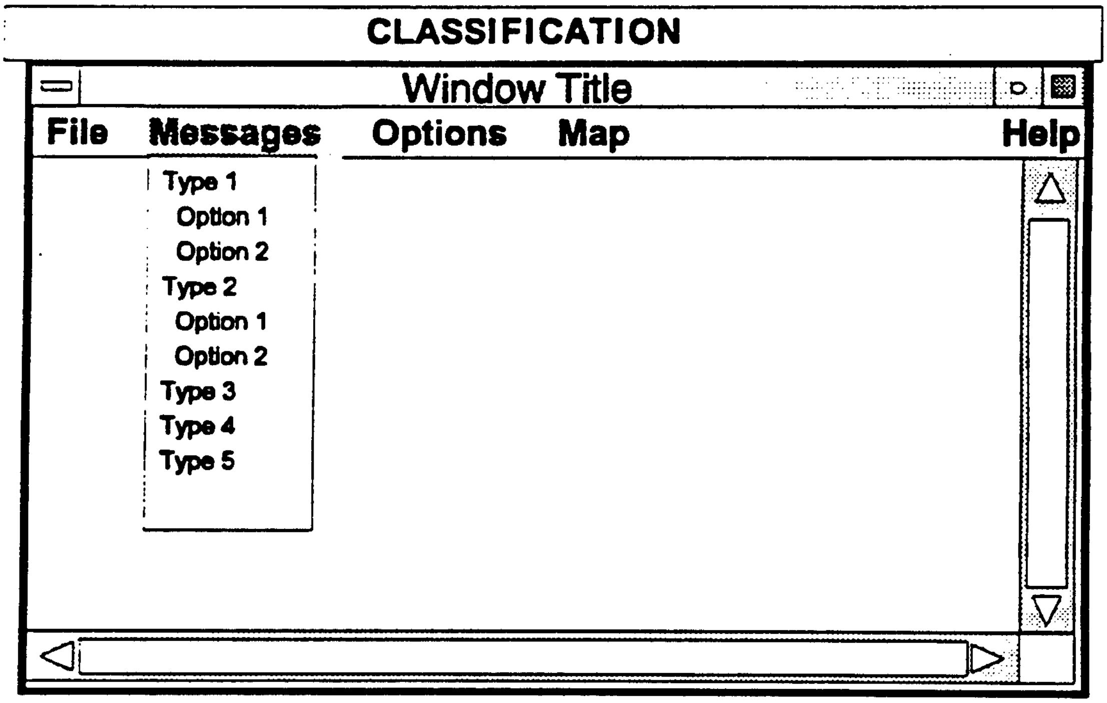


Figure 6-4. Example of Logical Ordering of Grouped Options

6.2.1.9 Single-Column List Format

When multiple menu options are displayed in a list, display each option on a new line (i.e., format the list as a single column).

6.2.1.10 Overlapping Items

Ensure that menu options do not overlap controlled functions or appear to do so to the user.

6.3 HIERARCHICAL MENUS

6.3.1 Usage

Use hierarchical menus:

- When menu selection must be made from a long list and not all options can be displayed at once
- If a selection list exceeds 10-15 items.

6.3.2 General Guidance

6.3.2.1 Organization and Labeling of Hierarchical Menus

When hierarchical menus are used, organize and label them to guide the user within the hierarchical structure. Identify currently active menu selections to the user. The preferred method is to use more than one mode (i.e., color and font, size and color of text, etc.).

6.3.2.2 Easy Selection of Important Options

Design hierarchical menus to permit immediate user access to critical or frequently selected options.

6.3.2.3 Indicating Current Position in Menu Structure

When hierarchical menus are used, display an indication of the user's current position in the menu structure. This could be done in the menu title, or as a page X of N notation on the menu page.

6.3.2.4 Consistent Design of Hierarchical Menus

When hierarchical menus are used, ensure the display format and option selection logic are consistent at every level of the hierarchical menu structure.

6.3.2.5 Graphic User Interface for Hierarchical Menus

Keep hierarchical menu design in a GUI as simple as possible. The use of complex graphic structures is distracting to the user.

6.3.3 Navigating Hierarchical Menus

6.3.3.1 Including a System-Level Menu

Provide a system-level menu of basic options as the top level in a hierarchical menu structure, as illustrated in Figure 6-5. The system-level menu will act as a home base to which a user can always return as a consistent starting point for control entries.

6.3.3.2 Organization and Labeling System-Level Menu Listed Options

Group, label, and order control options for the system-level menu in terms of their logical function, frequency, and criticality of use.

6.3.3.3 Return to the System-Level Menu

When hierarchical menus are used, require the user to take only one simple control action to return to the system-level menu.

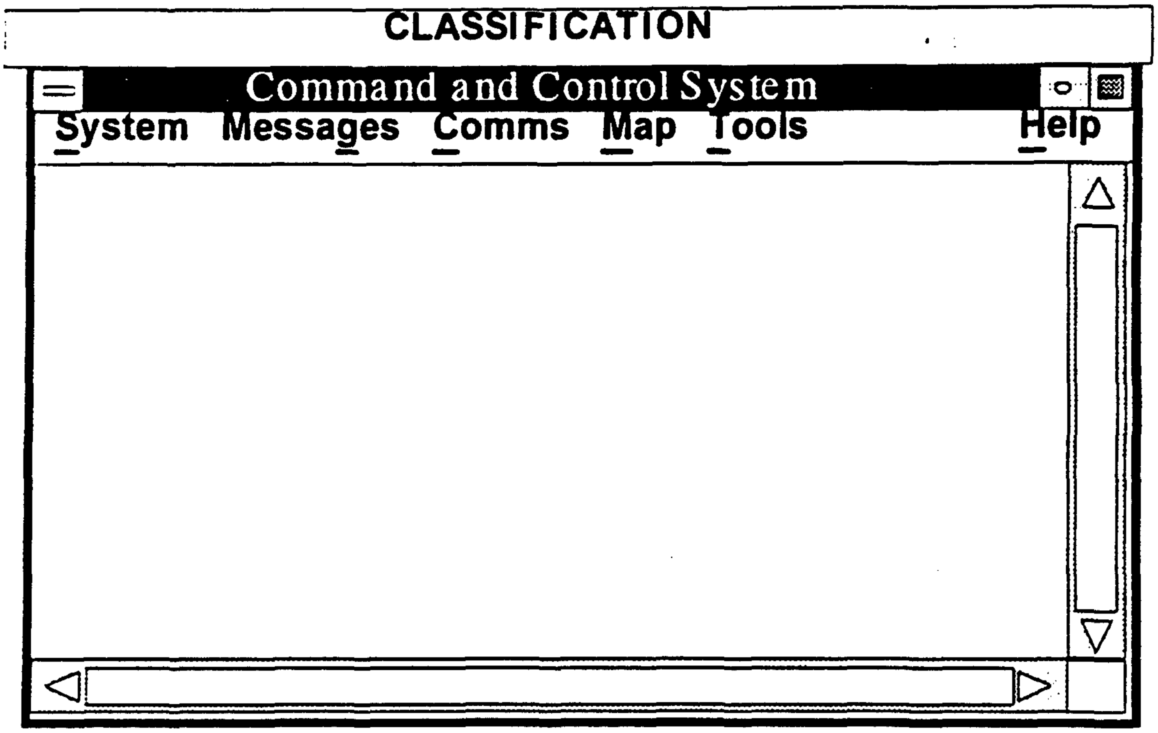


Figure 6-5. Example of a System-Level Menu

6.3.3.4 Return to Higher Level Menus

When hierarchical menus are used, require the user to take only one simple control action to return to the next higher level.

6.3.3.5 Control Options Distinct From Menu Branching

Format the display of hierarchical menus, dialog boxes, and pop-up windows such that options that actually accomplish control entries can be distinguished from those which merely branch to other menu frames. See Figure 6-6.

6.3.3.6 Hierarchical Menu-Browsing Methods in Direct Manipulation

Two basic methods for browsing options in hierarchical menus are used in direct manipulation interactive control: 1) select an option from one menu, which causes another menu to pop up, or 2) move the pointer towards the right side of an option, causing a menu to pop up (see Figure 6-7). It is recommended that both options be available.

6.3.3.7 Use of Multiple Paths

Provide multiple paths to accommodate both the experienced and inexperienced user. Allow the experienced user to use "type-ahead," "jump-ahead," or other shortcuts to navigate through the menu selection system.

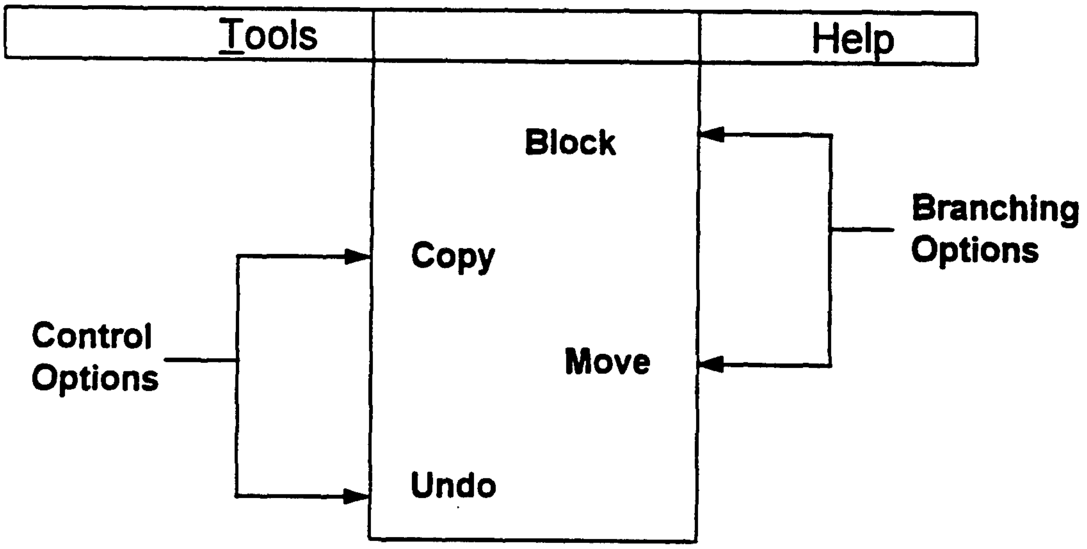


Figure 6-6. Distinction Between Control Options and Command Options

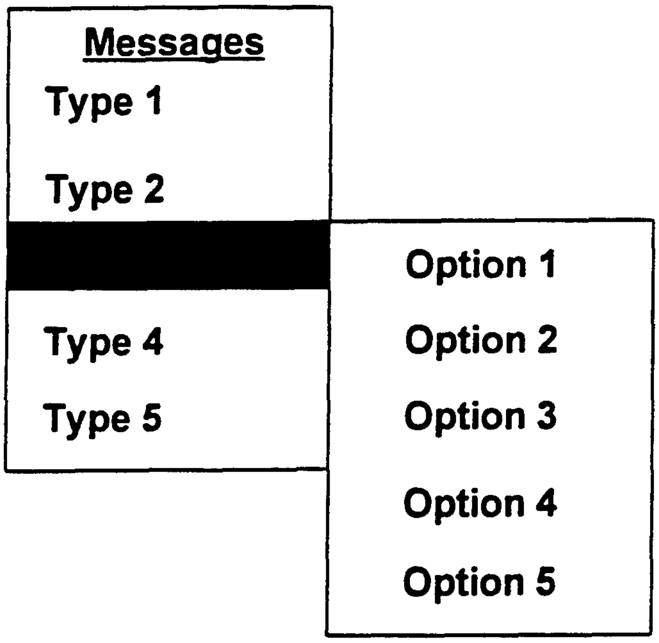


Figure 6-7. Example of a Hierarchical Menu

6.3.4 Hierarchical Menu Tree Depth and Breadth

6.3.4.1 Minimal Steps in Sequential Menu Selection

When the user must step through a sequence of menus to make a selection, design the hierarchical menu structure to minimize the number of steps required.

6.3.4.2 Use Broad Menu Trees

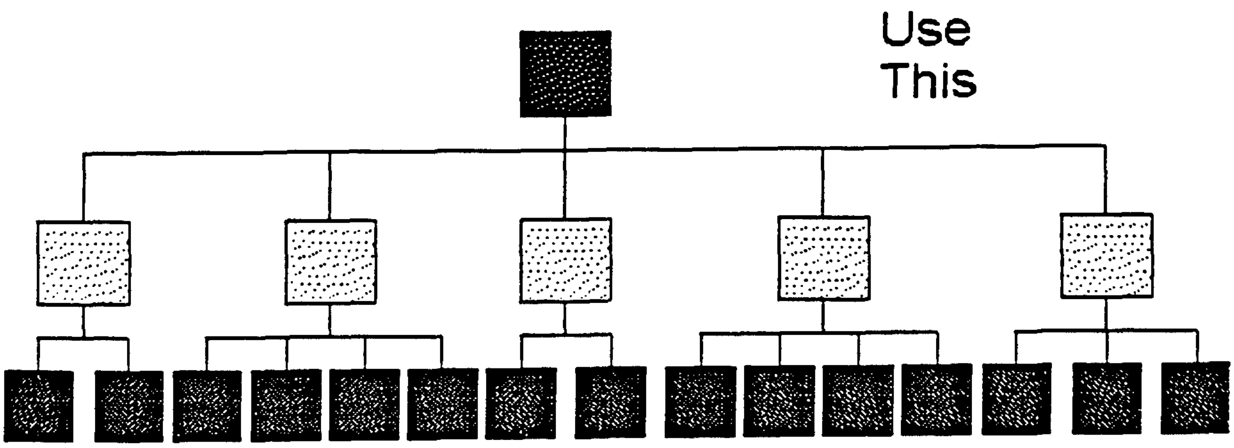
Use a broad and shallow menu tree, rather than a narrow and deep menu tree, for operational systems as illustrated in Figure 6-8.

6.3.4.3 Minimize Menu Choices in the Middle

Minimize the number of menu choices midway through a hierarchical menu, as the user is more likely to get lost at this stage.

6.3.4.4 Software Navigation Aids

Include in software navigation aids the ability to select a menu or submenu directly, without going through intermediate steps (see Figure 6-9). Enable the user to switch between software modules in a quick, easy manner, using an interface such as a tree or organization chart.



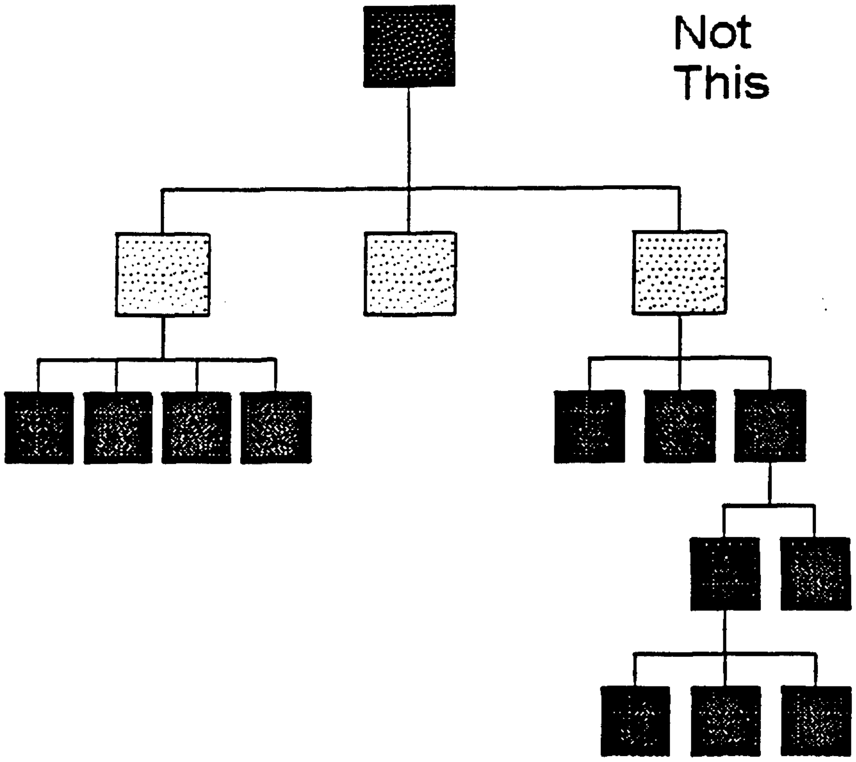


Figure 6-8. Broad and Shallow Menu Tree vs. Narrow and Deep Menu Tree

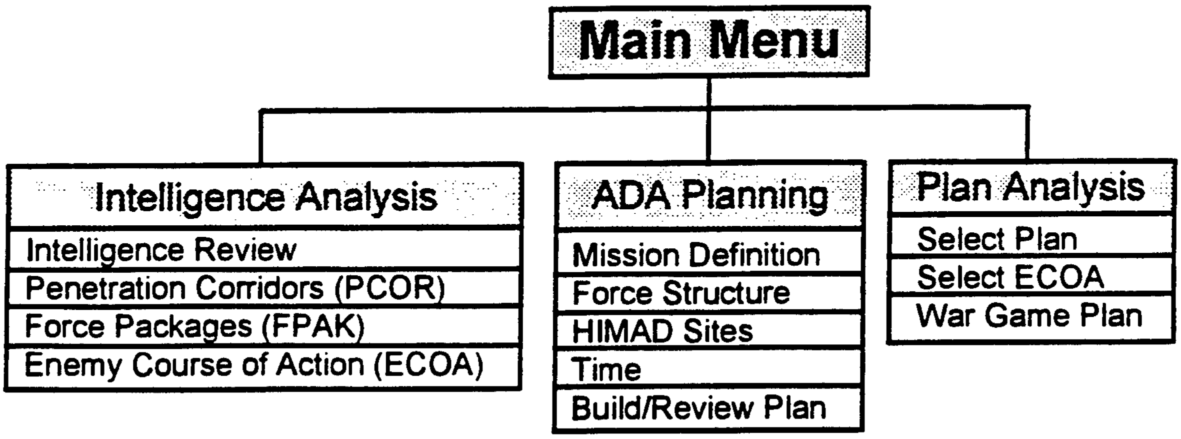


Figure 6-9. Example of a Tree Diagram Interface

6.4 ITEM SELECTION

6.4.1 General

6.4.1.1 Automatic Pointer Placement

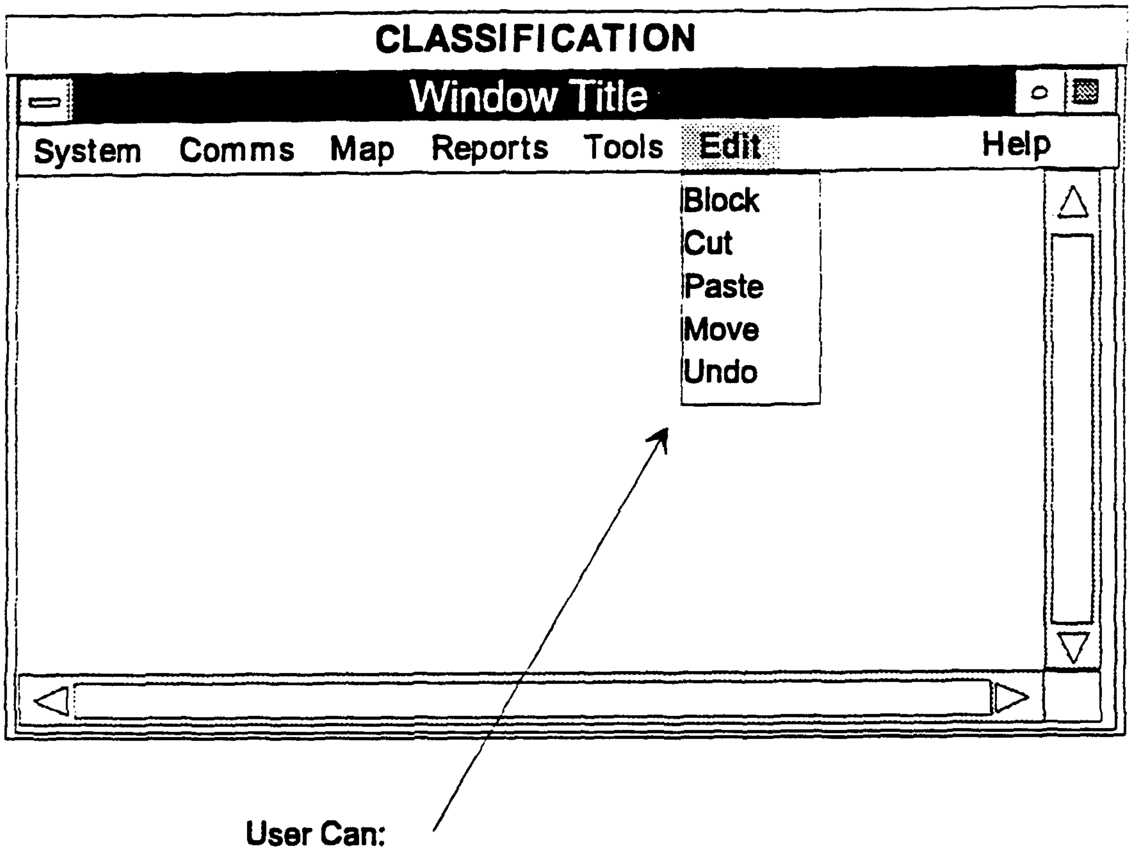
When pointing to make a menu selection on menu displays not included with data displays, ensure that the computer places the pointer automatically at the first listed option. When menu selection is by code entry, place the pointer in the command entry area.

6.4.1.2 Minimize Menu Selections

Keep the number of menu selections to the absolute minimum to reduce system menu-selection time.

6.4.1.3 Use a Combined Mode of User Interface

Enable users to use two modes for menu selection: keying in a numeric or letter code, or placing the pointer at the option and selecting (see Figure 6-10).



① Move pointer to highlight option and select, or
② Type key letter code

Figure 6-10. Example of a Combined Mode User Interface

6.4.1.4 Feedback for Menu Selection

When a user selects and enters a control option from a menu, if no natural response is immediately observable, the software should display some other acknowledgment of that entry. See examples in Figure 6-11. Where possible, the acknowledgment should be animated.

6.4.1.5 Standard Area for Code Entry

When menu selection is accomplished by code entry (other than mnemonics), provide a standard command entry area where the user enters the selected code. Place that entry area in a fixed location on all displays.

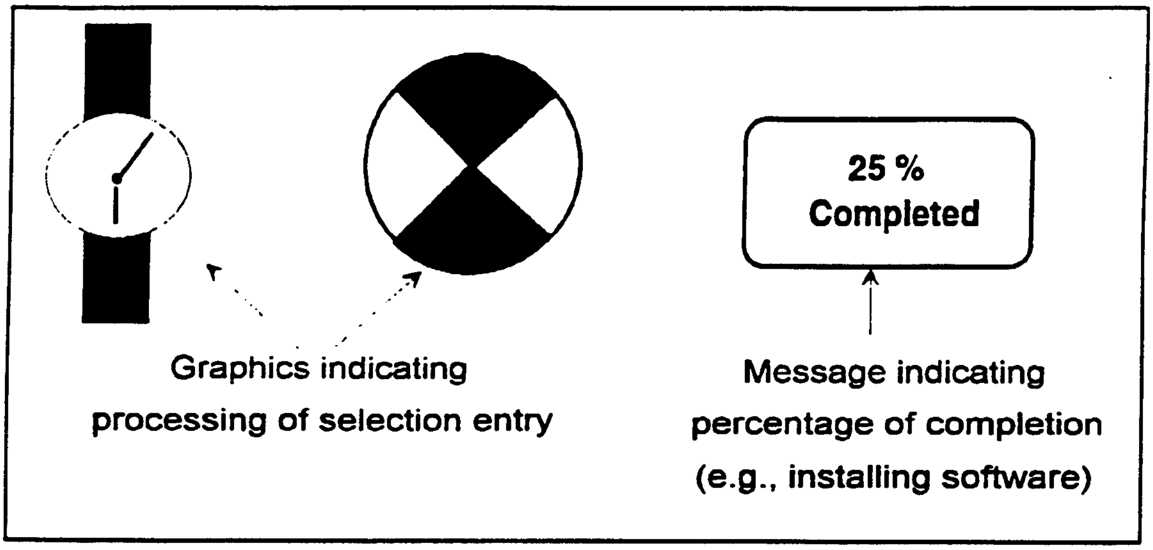

25 %
Completed
Graphics indicating processing of selection entry
Message indicating percentage of completion (e.g., installing software)

Figure 6-11. Graphic Acknowledgments of Selection-Processing

6.4.1.6 Allow Abbreviated Menu Selections

Allow menu selections by the user to be accepted in either abbreviated or complete form. For example, the user should be able to use Q, QU, or QUIT.

6.4.2 Selection By Pointing

6.4.2.1 Menu Selection by Pointing

If menu selection is the primary means of sequence control, and especially if choices must be made from extensive lists of displayed control options, permit option selection by direct pointing (e.g., mouse, trackball). See Subsection 7.1, general aspects of direct manipulation.

6.4.2.2 Large Pointing Area for Selecting Options

The acceptable pointing area for menu options should be as large as is consistently possible. Ensure that the area includes at least the displayed option label, plus a half-character distance around that label.

6.4.2.3 Dual Activation for Pointing

If pointing for menu selection, provide dual activation, where the first action designates (positions a cursor at) the selected option and a separate, second action makes an explicit control entry (e.g., clicking the mouse).

6.5 MENU OPTION LABELING

6.5.1 General

6.5.1.1 Use of Key Words

Ensure that menu items begin with a key word.

6.5.1.2 Menu Options Worded as Commands

Ensure that the wording of menu options consistently represents commands to the computer (e.g., File, Save, Edit), rather than questions to the user.

6.5.1.3 Menu Categories

Ensure that menu category labels are comprehensible and unique. The words, phrases, and titles should state options in clear English.

6.5.1.4 Labeling Grouped Options

If menu options are grouped in logical subunits, give each subunit a descriptive label distinctive in format from the option labels themselves (see Figure 6-12).

6.5.1.5 Use Familiar Terminology

Use familiar terminology when labeling menus, but ensure that items are distinct from one another.

6.5.2 Selector

6.5.2.1 Best and Worst Selectors for Menu Items

Mnemonics is a technique to assist in improving the user's memory. Compatible or mnemonic letters are the best selectors for menu items; incompatible letters are the worst. Numbers are intermediate selectors.

- Use lettered menu items if possible, as they have the following advantages: more single entry keys are available; there is less chance of a keying error; and mnemonic keying of entries is possible.

- Use numbered menu items as intermediate selectors, with the following advantages: sequencing of items is clear; non-typists can easily locate numbers; and the user can quickly see how many options are available.

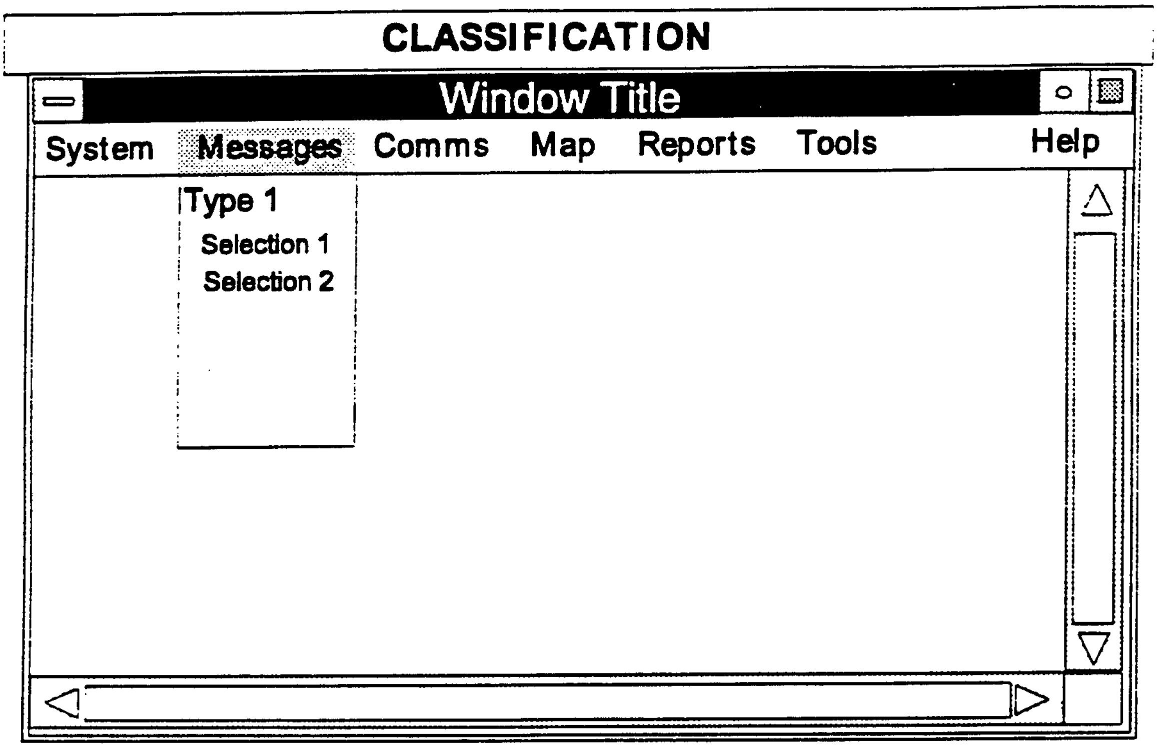


Figure 6-12. Example of Distinctive Subunit Labels

6.5.2.2 Do Not Combine Codes

Letter and numeric codes should not be combined in the dialog.

6.5.2.3 Selection of Menu Titles

Use selectors that closely match the item represented, to facilitate user retention of commands.

6.5.2.4 Numbering

Number menu items starting with 1 -- not with 0.

6.5.2.5 Consistent Coding of Menu Options

If letter codes are used for menu selection, use those letters consistently in designating options from one transaction to another.

6.5.2.6 Displaying Option Code

When the user must select options by code entry, display the code associated with each option in a consistent, distinctive manner, as shown below.

Code		Option
P	=	Previous Page
N	=	Next Page
U	=	Undo
Del	=	Delete

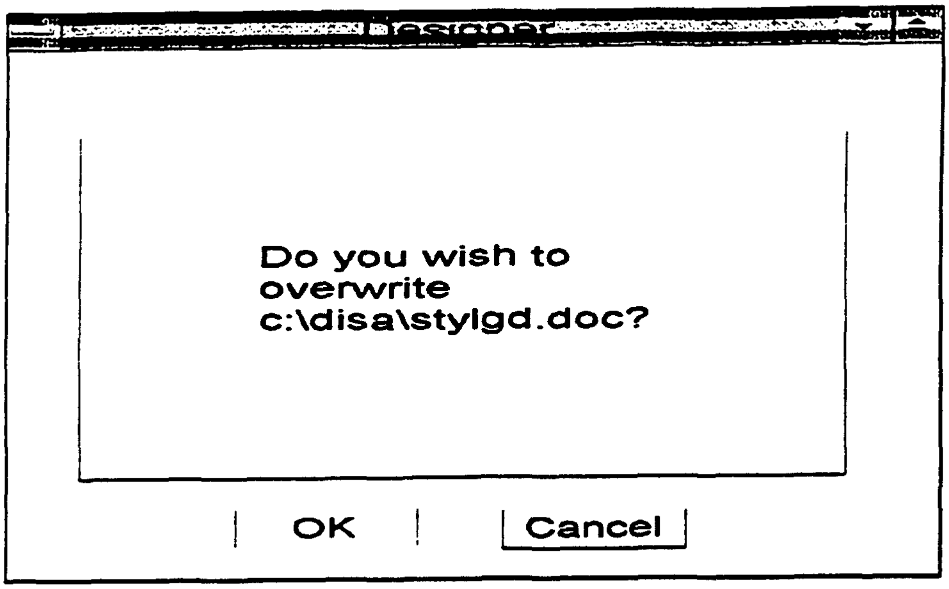
6.6 DIALOG BOXES/POP-UP WINDOWS

GUI style guides refer to windows that contain graphical controls (widgets), such as dialog boxes and pop-up windows (see Subsection 5.1, Window Basics), for interacting with applications. Examples of dialog boxes include message, question, warning, action, and command windows; examples of pop-up windows include command windows, property windows, and notices. Note that dialog box and pop-up command windows are not equivalent or even related. The former refers to a window that allows users to enter commands to the application or operating system, and the latter is a window that sets parameters and executes commands based on those parameters. These windows are used to:

- Display important messages or warnings
- Collect or solicit data from the user
- Modify and set properties of objects
- Notify the user of the progress of a lengthy process.

Dialog boxes and pop-up windows are invoked by applications in response to 1) user actions and requests, 2) unexpected or unplanned events (e.g., a printer runs out of paper), or 3) initiation of a time-consuming activity. See the dialog box in Figure 6-13. The application decides where and when they are displayed, but all dialog boxes and pop-up windows should include at least one button that solicits a response from the user. Windows should be noticeable but small and, if possible, moveable. It is recommended that only one dialog box or pop-up window be displayed at a time within any application in order to avoid clutter and confusion.

Dialog boxes and pop-up windows should automatically receive input focus. Users should be required to respond to dialog boxes or pop-up windows and should be prevented from returning the input focus to the main or primary window (of the application) until they have responded appropriately.



Do you wish to overwrite c:\disa\stylgd.doc?

Figure 6-13. Example of a Dialog Box

All types of dialog boxes and pop-up windows behave similarly, but they differ in content depending on the needs of the application. For example, a push button is always pushed and a check button is always checked, but each application will choose the types of controls to use and combine them differently. The following paragraphs provide recommendations for message wording and briefly describe some common types of dialog boxes and pop-up windows. More detailed descriptions can be found in the OAF/Motif and Open Look style guides.

6.6.1 Message Wording Guidelines

The following guidelines, which are designed to maximize user performance and accuracy, should be applied to dialog boxes, pop-up windows, message areas, and any other communications between the application and user.

- Use an abbreviation only when it is significantly shorter than the full word.
- Ensure that abbreviations are meaningful, recognizable, and used consistently.
- Do not abbreviate words not commonly abbreviated. For example, use "Restricted Acct No," not "Restr Account Number."
- Ensure that message lines end in full words rather than in hyphenations.
- Ensure that messages are directly usable, requiring no further documentation or translation.
- Avoid overly technical wording, and use short simple sentences that begin with the main topic.

- Avoid abrupt wording, such as INVALID, ILLEGAL, and FATAL.

- Focus error messages on the procedure for correcting the error, not on the action that caused the error.

- Display critical error messages (those requiring immediate response from the user to prevent invalid data or results) in caution/warning windows, as shown in Paragraph 6.6.4. Display noncritical messages in the message area at the bottom of the application window, as described in Paragraph 5.1.5.

- Where appropriate, the HELP facility can be used to expand more fully on messages.

6.6.2 Work-In-Progress Window

When a user's request is simple and requires five seconds or less processing time, feedback can be in the form of a changed pointer shape or a brief message within the window. When the request exceeds five seconds, the application should provide a work-in-progress window to indicate a time-consuming operation is taking place. If appropriate, provide a means by which the operation can be canceled or aborted. The application removes the box when the operation has been completed.

Ensure that the application shows the status of the operation by a dynamically changing progress indicator (e.g., "10% Sorted," "4 out of 10 files copied," or a scale showing status).

6.6.3 Information Box

An application should generate an information box (i.e., a Motif message box or an Open Look notice) when the application needs to display an information message. This window should be reserved for noncritical messages requiring acknowledgment by the user. An application's frequent informational messages should be displayed in the window's message area (see Paragraph 5.1.5).

An information box can freeze the application and require the user to explicitly dismiss the window before proceeding. If the halted operation can be retried, include a "Retry" button within the message window. If a default push button is designated, assume it is the desired action.

6.6.4 Caution/Warning Box

A caution/warning box, containing critical messages that warn the user of the consequences of carrying out an action, usually includes "Yes," "No," and Cancel" buttons. The message should be an unambiguous question or statement. When this box is displayed, suspend the application until the user provides instructions on how to proceed. Ensure that the default push button is always the least destructive operation.

6.6.5 Menu Box

A menu box is the result of the user's selecting a routing or window menu item. Menu boxes solicit data from users through a combination of controls (e.g., entry boxes and settings). Name the menu box in accordance with the menu item that created it. For example, the "search..." menu item should generate a menu with the title "Search...." A "Cancel" push button should be included in the window to allow users to close the menu box. If a default push button is designated, it should be the assumed desired action.

REFERENCES

Paragraph	References
6.1.1	Shneiderman (1987) p. 107
6.1.2	Shneiderman (1987) p. 115
6.1.3	Smith and Mosier (1986) para 3.1.3-16; Sidorsky (1984) para 6.1-14
6.1.4	Smith and Mosier (1986) para 3.1.3-36
6.1.5	Smith and Mosier (1986) para 3.1.3-20
6.1.6.1	Ziegler and Fähnrich (1988) p. 130
6.1.6.2	Ziegler and Fähnrich (1988) p. 130
6.1.7	Ziegler and Fähnrich (1988) p. 130
6.1.8.1	Ziegler and Fähnrich (1988) p. 129
6.1.8.2	Ziegler and Fähnrich (1988) p. 129
6.1.8.3	Ziegler and Fähnrich (1988) p. 129
6.2.1.1	Lickteig (1989) p. 13, 30 Appendix A p. A-1; Bailey (1982) p. 346; Chao (1986) p. 15
6.2.1.2	Smith and Mosier (1986) para 3.1.3-19; Shneiderman (1988) p. 702.
6.2.1.3	Smith and Mosier (1986) para 3.1.3-22 and 4.4-3; Shneiderman (1988) p.702; Paap and Roske-Hofstrand (1986) p. 384
6.2.1.4	Smith and Mosier (1986) para 3.1.3-23; Lickteig (1989) p. 15; Shneiderman (1988) p. 702
6.2.1.5	Galitz (1984) p. 120
6.2.1.6	Galitz (1984) p. 120; Chao (1986) p. 16
6.2.1.7	Williams et al. (1987a) Appendix p. A-3
6.2.1.8	Smith and Mosier (1986) para 3.1.3-17 and 3.1.3-18; Sidorsky (1984) para 6.1-13
6.2.1.9	Smith and Mosier (1986) para 3.1.3-3
6.2.1.10	Shneiderman (1987) p. 100
6.3.1a	Smith and Mosier (1986) para 3.1.3-25; Nielsen (1987) p. 384; Lickteig (1989) p. 13; Ziegler and Fähnrich (1988) p. 129

REFERENCES (Cont.)

Paragraph	References
6.3.1b	Sidorsky (1984) para 6.1-14; Chao (1986) p. 15
6.3.2.1	Bowser (1991) p. 9; Smith and Mosier (1986) para 4.4-4
6.3.2.2	Smith and Mosier (1986) para 3.1.3-28; Sidorsky (1984) para 5.1-14; Galitz (1984) p. 119
6.3.2.3	Smith and Mosier (1986) para 3.1.3-30; DoD (1989a) p. 267; Chao (1986) p. 15; Shneiderman (1987) p. 115
6.3.2.4	Smith and Mosier (1986) para 3.1.3-32
6.3.2.5	Bowser (1991) p. 9
6.3.3.1	Smith and Mosier (1986) para 3.1.3-26, 4.4-2, 3.2-2; Shneiderman (1988) p. 702; Galitz (1984) p. 120; Sidorsky (1984) para 1.1-12, 3.2-21
6.3.3.2	Smith and Mosier (1986) para 3.2-3
6.3.3.3	Smith and Mosier (1986) para 3.1.3-34; Galitz (1984) p. 120; DoD (1989a) p. 266
6.3.3.4	Bowser (1991) p. 10; Smith and Mosier (1986) para 3.1.3-33; Sidorsky (1984) para 5.1-14; DoD (1989a) p. 267; Chao (1986) p.16
6.3.3.5	Smith and Mosier (1986) para 3.1.3-31; Galitz (1984) p. 120
6.3.3.6	Ziegler and Fähnrich (1988) p. 129
6.3.3.7	Smith and Mosier (1986) para 3.1.3-35; Shneiderman (1988) p.702; Shneiderman (1987) p. 118; Chao (1986) p. 16; Laverson and Shneiderman (1987) p. 104; Sidorsky (1984) para 5.1-15
6.3.4.1	Smith and Mosier (1986) para 3.1.3-27; Sidorsky (1984) para 5.1-14
6.3.4.2	Shneiderman (1988) p. 702; Norman and Chin (1988) p. 63
6.3.4.3	Norman and Chin (1988) p. 63
6.3.4.4	Galitz (1984) p. 120
6.4.1.1	Galitz (1984) para 3.1.3-29; Chao (1986) p. 15
6.4.1.2	Parkinson et al. (1988) p. 691
6.4.1.3	Antin (1988) p. 181

REFERENCES (Cont.)

Paragraph	References
6.4.1.4	Smith and Mosier (1986) para 3.1.3-9
6.4.1.5	Smith and Mosier (1986) para 3.1.3-8; Chao (1986) p. 15
6.4.1.6	Sidorsky (1984) para 6.1-15
6.4.2.1	Smith and Mosier (1986) para 3.1.3-4; DoD (1989a) p. 266
6.4.2.2	Smith and Mosier (1986) para 3.1.3-5
6.4.2.3	Smith and Mosier (1986) para 3.1.3-6
6.5.1.1	Shneiderman (1988) p. 702
6.5.1.2	Smith and Mosier (1986) para 3.1.3-11; Sidorsky (1984) para 5.1-13
6.5.1.3	Shneiderman (1987) p. 87; Bailey (1982) p. 343
6.5.1.4	Smith and Mosier (1986) para 3.1.3-24; Sidorsky (1984) para 5.1-14; Chao (1986) p.16
6.5.1.5	Shneiderman (1987) p. 100
6.5.2.1	Laverson et al. (1987) p. 106; Shneiderman (1987) p. 117 and 118; Chao (1986) p. 16; Smith and Mosier (1986) para 3.1.3-13; Galitz (1984) p. 120
6.5.2.2	Chao (1986) p. 16
6.5.2.3	Laverson et al. (1987) p. 105
6.5.2.4	Sidorsky (1984) para 2.1-13; Chao (1986) p. 16
6.5.2.5	Smith and Mosier (1986) para 3.1.3-14; Sidorsky para 5.1-14
6.5.2.6	Smith and Mosier (1986) para 3.2-8; Shneiderman (1987) p. 115
6.6	DISA/CIM (1992b)

This page intentionally left blank.

7.0 DIRECT MANIPULATION

Direct manipulation is the major type of interactive dialog for GUIs. In a direct manipulation dialog, the user controls the interface with the computer by acting directly on "objects" on the display screen. This object may be an icon, menu option, symbol, button, or dialog box. The user highlights the object and implements the action by using a pointing device, such as a mouse or trackball. Sample actions include moving an object, querying a database, calling up a preformatted message template, or sending a message over a communications system. The result of an action is immediately observable.

Direct manipulation of a computer system is analogous to controlling a vehicle. The user uses a control, such as the steering wheel, to input a command to the vehicle and is rewarded by an immediate response. With a computer, the user moves a pointer over an object, such as an icon, and presses a pointing device control (i.e., button) to input a command, such as querying status. The direct manipulation system responds immediately by displaying the status in a pop-up window next to the object. In contrast, when using a command-language-based system, the user types in a command, hits ENTER, then waits for a response from the system.

Direct manipulation user interfaces are characterized by continuous representation of the object of interest and by computer actions accomplished by physical actions such as button presses, incremental reversible actions, and immediate visual feedback. These characteristics provide the user with a greater feeling of control and often result in better performance and greater acceptance of the system.

Direct manipulation in the user interface reduces the time required to learn new applications. Efficiencies in learning result from using both standard, consistent actions in the application environment and metaphors to guide the user. A metaphor uses the visual nature of a direct manipulation interface to map objects in the application onto a visual representation familiar to the user.

Metaphors effectively control the complexity of the user interface because they make actions, procedures, and concepts similar to those already known to the user. By capitalizing on the user's prior knowledge, the designer permits the user to think in terms familiar to the application domain, rather than in terms of low-level computer concepts. The resulting applications are easier to learn and easier to use. An effective example is the office metaphor associated with the Apple-MacIntosh interface style.

The following pages provide detailed guidelines for designing the user-computer interface when direct manipulation is used. The designer should recognize that the literature has little explicit guidance for direct manipulation as it applies to specific military systems. Therefore, it is imperative that before formalizing the user-computer interface, detailed research and careful testing of alternatives be done with users who represent the intended user population.

7.1 GENERAL

Provide direct manipulation of displayed objects as a means of interactive control. Direct manipulation works particularly well for applications where there will be many casual system users and where turnover in personnel will be high, such as in operational military situations.

7.1.1 Hardware Considerations

The designer should consider the following hardware factors for an effective direct manipulation system:

- Use high-resolution screens and a bitmapped hardware architecture, as these are required for direct manipulation systems. The bitmapped windowing system requires greater central processing and memory size as well as rapid operating speed to provide the immediate response for effective user-computer interaction.

- Because direct manipulation is designed to represent the actual product, a positive image (dark foreground on light background) is best, as it represents printed output. See the example in Figure 7-1.

- Direct manipulation is most efficient when using a pointing device, such as a mouse, trackball, or touch-interactive device. Section 3.0 discusses touch-interactive devices. Software must be flexible enough to accommodate keyboard cursor keys or accelerators should the pointing device fail.

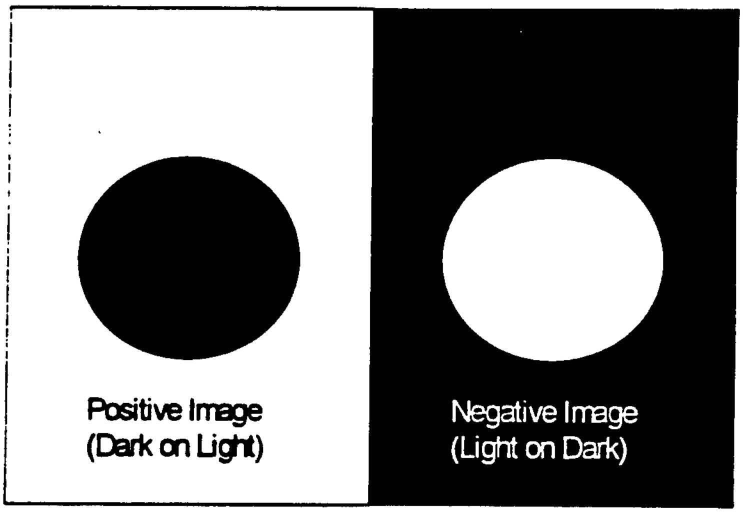


Figure 7-1. Example of Positive and Negative Images

7.1.2 Screen Arrangement by the User

Enable the user to arrange windows and icons on the screen to meet the individual task needs. However, do not allow the user to move a window or icon to a nonretrievable position (i.e., off the screen).

7.1.3 Function Control

Five methods should be considered for invoking a function, file, or operation with direct manipulation: Function Keys, Menu Bar, Pop-up Menus, Pull-down Menus, and Icons. Icons are discussed in Subsection 7.3, function keys are discussed in Subsection 8.4, and menuing is discussed in Section 6.0.

7.1.4 Interaction

Operator interactive tasks should use the most appropriate input mode. The keyboard is recommended for extensive alphanumeric data. It is usually more effective to use pointing devices to select from menus. Where both modes are present, the software should allow both keyboard and pointing device selection of items. Operational military systems should provide complete inter-changeability of keyboard and pointing device for emergency situations.

7.2 METAPHORS

Metaphors associate interface objects in the application or functionality with a visual representation familiar to the user. To capitalize on information-carrying capacity, use metaphors to unify individual icons into integrated groupings using established attributes and associations of real-world objects. Icons replace commands and menus as the means to support end-user dialog (e.g., trash can replaces the delete command or menu item).

In investigating the range of understandability of symbols, research indicates the need to evaluate symbols and icons before their widespread adoption. The metaphor used for icons and system interaction should be tested in advance with representatives of the intended user population. Some symbols have very little meaning, can be misleading, and can cause potentially dangerous confusion in international environments.

Each society has unique meanings for different types of graphics and icons. These meanings have been developed in the cultural evolution process by associating objects in natural and man-made environments with life events and activities within the society. Therefore, icons and graphics of each culture can and should be studied not only intraculturally but also cross-culturally. Intercultural learning can be facilitated through formal education and cultural assimilation training of the semantic features involved in verbal and nonverbal communication.

7.2.1 Metaphor Selection

Understandability remains of primary concern in achieving effective and widely-accepted symbols. The following paragraphs provide guidance on metaphor selection and design.

7.2.1.1 System Model

The metaphor selected for icon design should model the system being controlled.

7.2.1.2 Appropriate to Task

The metaphor selected for icon design should be appropriate for the user's tasks, functions, and environment (e.g., the office metaphor may not be appropriate for some military applications).

7.2.1.3 Leveraging Knowledge

Selecting the metaphor should leverage prior knowledge in a way that is specific to the user environment.

7.2.1.4 Functional Representation

The metaphor should represent the system function in a way that is meaningful to the user.

7.2.1.5 Generalization of Metaphors

Metaphors should be general enough to allow the user to understand and use other metaphors or media, such as text-based systems.

7.2.2 Metaphor Design

7.2.2.1 Complex Metaphors

Avoid using complex metaphors. Complex metaphors, like complex icons, can lead to increased inferences of meaning and errors by the user. For example, the metaphor of biological evolution, if used to describe the levels and layers of an application, would be an overly complex metaphor.

7.2.2.2 Metaphor Oversimplification

Although metaphors should be as simple as possible, avoid oversimplification. Oversimplification occurs when the metaphor does not model full capability of the system. This oversimplification can cause underutilization of the system functionality. For example, the metaphor of a notepad, if used to describe the levels and layers of an application, would be an overly simple metaphor.

7.2.2.3 Metaphor Consistency with Objects

Metaphors should be consistent with the objects chosen to represent the functions. For example, deleting a file with recovery capability would be represented by a trash can, whereas deleting a file permanently would be represented by a paper shredder.

7.2.2.4 Metaphors Versus Self-Contained Icons

If effective self-contained symbols (icons) can be designed for information presentation, use them over the multiple icons of a complex metaphor.

7.2.2.5 Metaphor Tutoring

Design icon metaphors to tutor the user towards a more complete understanding of the underlying functional system.

7.2.2.6 Connotations Induced by Metaphors

Develop metaphors carefully, especially those used by more than one cultural or national group (e.g., NATO forces). Ensure that metaphors do not have a negative connotation for the user. For example, the "OK" sign formed by touching the forefinger tip to the thumb tip carries obscene connotations for some cultures.

7.3 ICONS

Icons are pictographic symbols representing underlying objects, concepts, processes, or data in a computer system. Icons are visible manifestations of a metaphor. Basic principles for designing icon and symbol systems are similar to those for designing large-scale windows and screens. Consistency, clarity, simplicity, and familiarity are key attributes. Sometimes these factors are at cross-purposes and require weighing one factor more heavily than another.

Visual communication, including symbols and icons, has three distinct interrelated dimensions: semantic, syntactic, and pragmatic. The strengths of icon design can be evaluated in terms of these basic components of communication. The semantic dimension refers to the relationship of a visual image to a meaning. Syntactic dimension refers to the relationship of one visual image to another, and pragmatic dimension refers to the relationship of a visual image to a user.

Icon images should be designed to meet unique communication needs, while maintaining a visual consistency throughout, using constant scale and limited size variations, orientation of figures with respect to text, use of colors, variation of line weights, and treatment of borders. These visual themes establish recognizability, clarity, and consistency while avoiding unnecessary variation of curves, line thickness, shape, color, and number of parts.

All icons, simple and complex, must function as a group with a recognizable visual vocabulary. Simplifying the images and amount of detail (e.g., eliminating unimportant features) results in consistently bold and direct symbols. Using an optically consistent line weight creates unity. Softening the edges (e.g., with curves) establishes visual relationships throughout the group.

It is beneficial to incorporate testing procedures as integral parts of the symbol development process, and not solely as a post-design evaluation. Criteria other than understandability also require consideration for many applications. For example, in an operational setting, the ability to distinguish the icon during over-the-shoulder (supervisory) viewing should be considered.

7.3.1 Types of Icons

Icons and symbols are most effective when they represent a service or concession that can be represented by an object and less effective when used to represent a process, activity, or complex interactions. Even an experienced user may become confused when trying to deal with large numbers of arbitrary icons. Four basic icon types are defined below and illustrated in Figure 7-2.

- **Resemblance** - Depict the underlying referent through an analogous image. The road sign is a good example.
- **Exemplar** - Serves as a typical example for a general class of objects. The knife and fork is used to represent restaurant services. This simple image is a very powerful depiction of the salient attributes associated with what one does in a restaurant.
- **Symbolic** - Convey the underlying referent at a higher level of abstraction than the image itself. The wine glass effectively depicts the abstract concept of "fragility."
- **Arbitrary** - Bears no relationship to the referent and therefore the association must be learned. The biohazard sign is a typical example of the arbitrary form where no physical or analogous correspondence exists between the symbol or icon and the intended meaning.

7.3.2 Icon Usage

When using direct manipulation, use icons as visual representations of system functions available to the user. The larger the symbol set the more difficult it will be to learn the symbols.

7.3.2.1 Iconic Menus

Iconic menus are groups of icons that act the same as textual menus, allowing selection of system options. Figure 7-3 illustrates an iconic menu.

- When users do not share a common language (e.g., NATO Forces), devise iconic menus for control functions. Test to be sure icon/text label is appropriate to many cultures. Examples from international signage may be appropriate.
- Place a limit on the number of icons shown at one time.

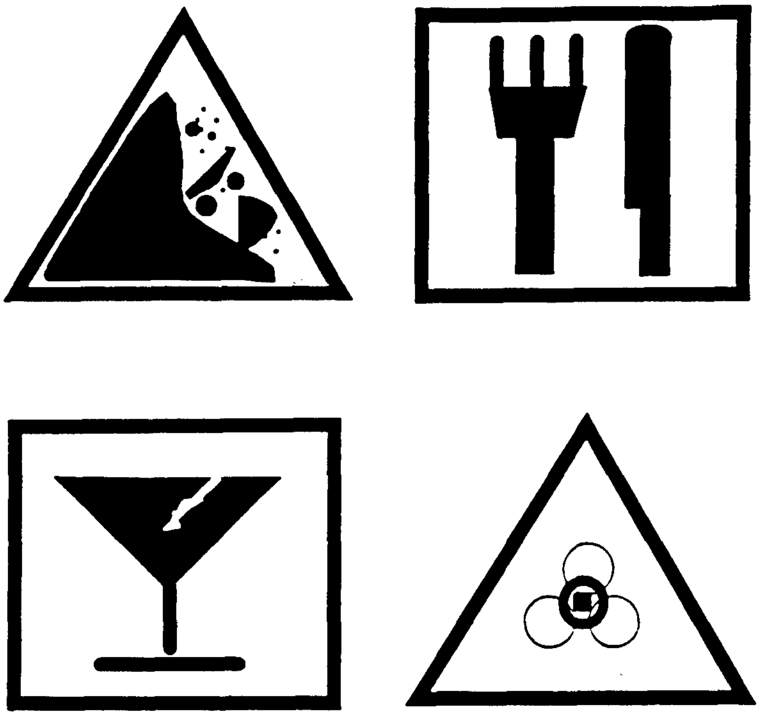


Figure 7-2. Examples of Four Basic Icon Types

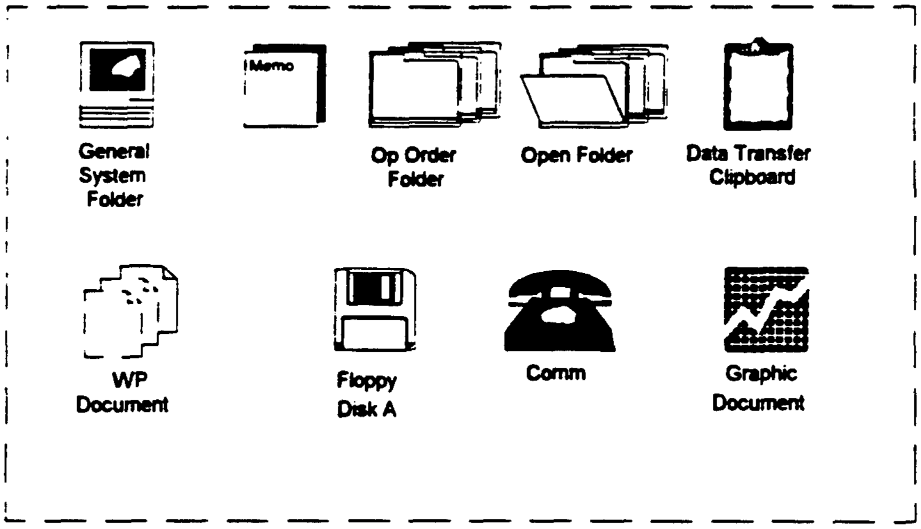


Figure 7-3. Example of an Iconic Menu

- Divide the screen display into cells (grids) capable of holding one icon. Well ordered menu locations increase predictability and consistency as well as decrease clutter.

- Provide a consistent location for icons (i.e., frequently used icons should be positioned in the edges), but allow the user to customize locations during use.

- Ensure icon sizing and location are consistent with other aspects of the design (e.g., windows).

- Use existing icons when available.

- When using iconic menus, design the system such that once an action has been initiated through an icon (e.g., printing), nonselectable icons cannot be manipulated. Provide the user with a visual indication of which icons are unavailable (e.g., dimmed/shadowed appearance when unavailable).

- Highlight the icon when it is selected.

7.3.2.2 Command Icons

Command icons are computer icons representing frequently used computer commands and operations. Apply general design principles for icons to command icons.

- To the extent possible, ensure that command icons are standardized and consistent across all DoD applications (e.g., common set of icons for command and utility functions within tactical/operational applications).

- The greater the risk or danger, the more standardized the icon should be.

- Ensure that command icon meaning/function is consistent across displays and standardized within an application.

- COTS software must meet consistency requirements for icon use and design within an application (e.g., when COTS applications use metaphor/icon designs inconsistent the with suggested universal icon approach, ensure consistency within COTS software itself).

7.3.2.3 Button Layout

Horizontal or vertical layout of buttons is a new use of icons in COTS such as word processing packages. This feature provides quick access to frequently used commands and macros. The buttons on the bar perform the commands directly when selected with a pointing device. See example in Figure 7-4.

- Button layouts should be customizable. The user should be able to select functions and macros for inclusion on the button layout.

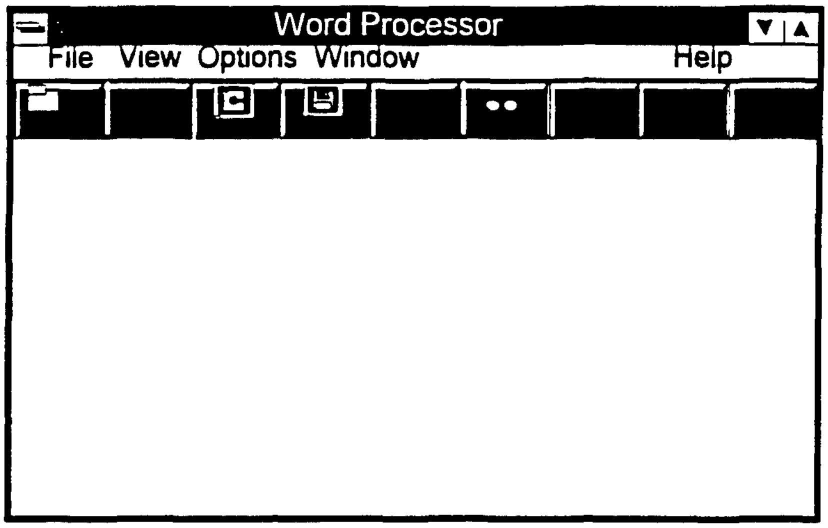


Figure 7-4. Example of Icons Used on Button Layout

- Design should allow selection with keystrokes as well as pointing device.
- Provide the capability to display and hide button layouts.

7.3.2.4 Icon Mapping

Iconic mapping implies the extent to which the link between the functionality depicted in the icon and the underlying functionality can be inferred. When an icon represents a group of functions (i.e., one to many mapping), do not repeat the specific icon for each function. Selecting the icon should cause an interface to appear that allows the user to select the specific function to be performed. For example, an icon for selecting communication devices, when selected, would bring up a window of types of communication devices available. The user would then select the appropriate device (see Figure 7-5).

7.3.2.5 Switching to Textual Representation

Include a feature that allows the user, when working with icons, to switch to a textual representation of the functions or files. The text should be listed sequentially to produce a logical transition from icon to text (see Figure 7-6).

7.3.3 Icon Design

Basic principles for icon design are similar to those for designing large-scale windows and screens. In one survey, participants were asked to rank icons from most to least appropriate. The results are outlined below:

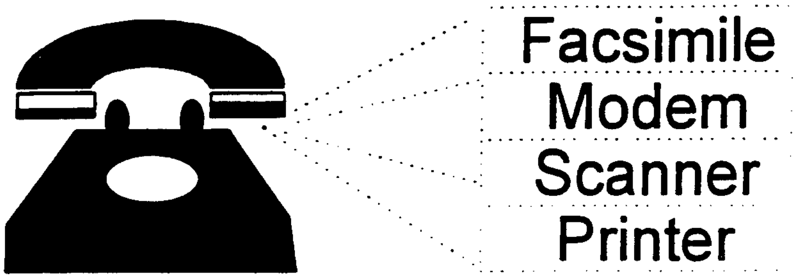
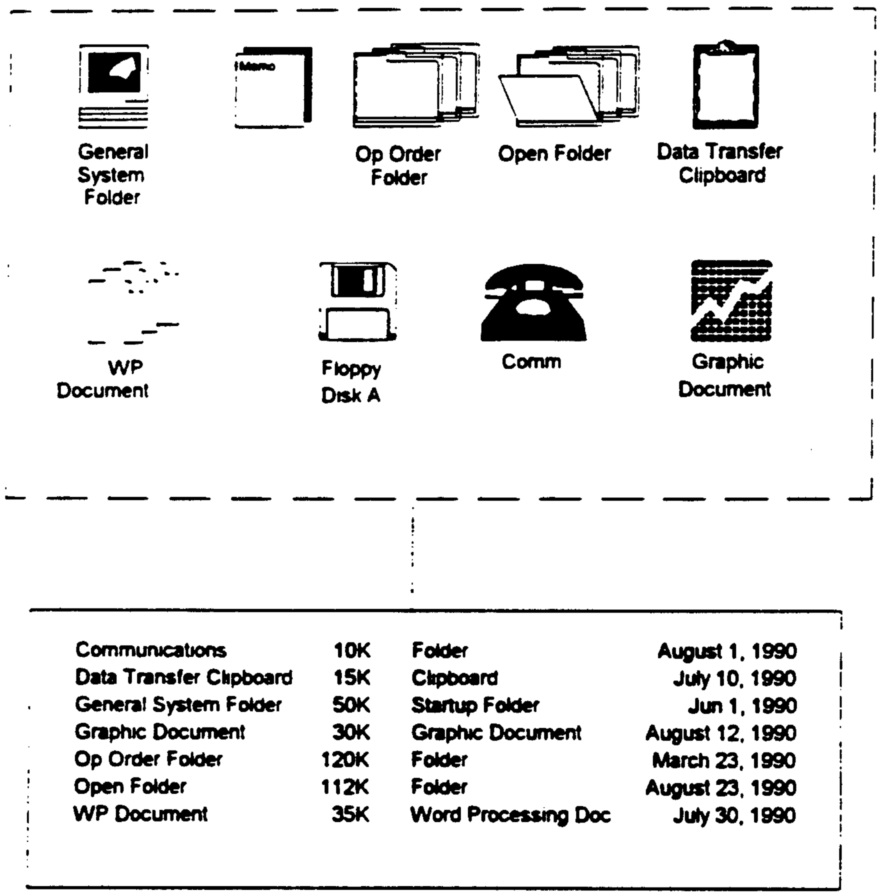


Figure 7-5. Example of a Multi-Function Icon with Interface for Selection of Available Functions



Communications	10K	Folder	August 1, 1990
Data Transfer Clipboard	15K	Clipboard	July 10, 1990
General System Folder	50K	Startup Folder	Jun 1, 1990
Graphic Document	30K	Graphic Document	August 12, 1990
Op Order Folder	120K	Folder	March 23, 1990
Open Folder	112K	Folder	August 23, 1990
WP Document	35K	Word Processing Doc	July 30, 1990

Figure 7-6. Example of Textual Representation

- Respondents preferred more concrete icons because it was easier to make association with something familiar. European respondents preferred more abstract icons, presumably because, in many everyday environments, they were more familiar with pictographic representations.

- Criticisms included that some icons were too similar and too numerous. Perceived similarity is a design concern because the ability to discriminate icons is an important feature.

- Some respondents asked for a box around all icons, while others disagreed. However, a box makes visual discrimination more difficult, especially for icons with image content near the perimeter of the box.

- Respondents requested a clearer indication of which activities (types or metaphors) should be represented by icons.

The designer should recognize that the ways in which a human perceives figures affects how icons are designed. The icon designer needs to incorporate Gestalt principles of human perception, briefly described as follows:

- Humans see the simplest or most efficient interpretation of an icon.

- The user will associate a meaning with an icon.

- Users tend to mentally group objects.

- Figure-ground relationships are important to how a user perceives an icon.

7.3.3.1 Consistent Icon Design

- Icon meaning should be consistent across displays and standardized within an application. To the extent possible, it should also be standardized and consistent across all DoD applications.

- As feasible, use a common set of primitives (software code that defines a specific shape, form, or color) and boundaries for icons. This will improve the user's ability to recognize and associate icons with their meanings.

- Ensure that icon is distinguishable from other icons (e.g., it shouldn't be similar to or confused with others).

- Ensure the icon can be seen well from all angles.

- In general, read icon pictures as books in the western culture, from top to bottom, left to right.

- Users prefer concrete symbols over abstract ones and simplicity over complexity.

- Design must avoid ambiguity. Ensure that no more than one meaning can be attributed to an icon.

- Use care when transferring design principles from one environment to another.

- Ensure that the user cannot select an invalid icon choice. Make unselectable those icons that should not be selected, and provide visual indication of this to the user.

- As a minimum, ensure that every icon includes the attributes of color, location, and visibility.

- Use existing icons that the user will recognize.

- Where possible, do not use icons unique to an application.

- Avoid the use of company logos and other such unique icons.

7.3.3.2 Use of Color

- Research shows that color offers no special advantage in speed of recognition. It is recommended that icon design be based on black and white rather than on color.

- Use caution when color-coding, and use color only if it is redundant to another coding method. See Subsection 4.3 for additional color guidelines.

- Use color with discretion. Too much color variation will confuse the viewer by creating clutter.

- In general, for color display, use five or fewer colors including black, white, and/or gray for icons.

- Use simple color patterns for background or low light areas.

- Limit colors to a carefully chosen set, and use them consistently across content areas and different display media.

7.3.3.3 Icon Labeling

- An icon shape generally represents a class of items or functions. To distinguish the precise function, provide a text label that names each icon. Pictures are remembered better if they have been named.

- Place Icon labels underneath the icon, as illustrated in Figure 7-7. If labels are not used, the user should be able to query the system to get a definition of the icon.

- Keep textual material simple in icon labels.

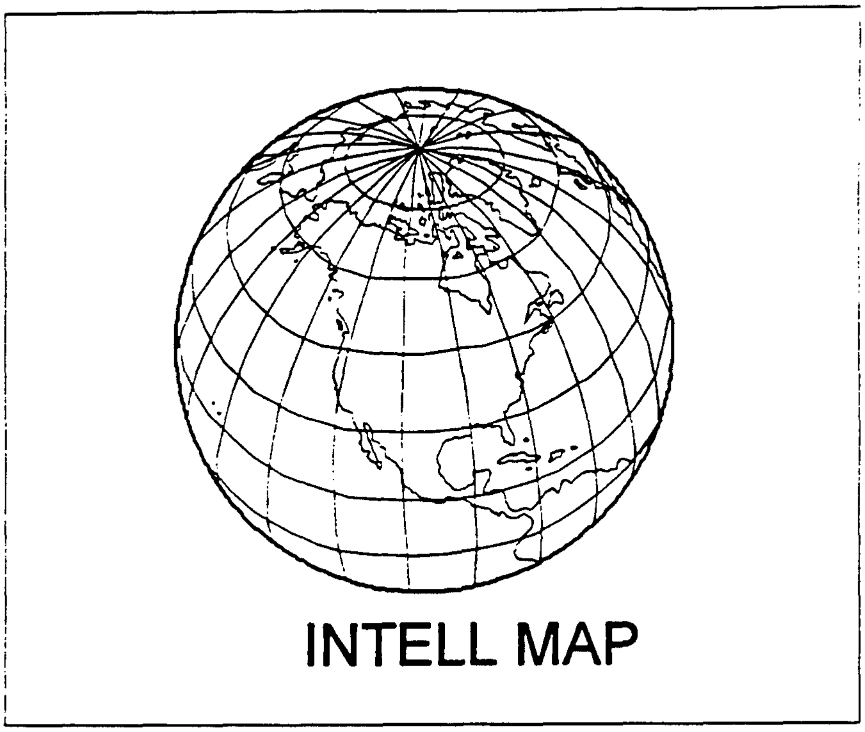


Figure 7-7. Example of a Text Title for an Icon

7.3.3.4 Icon Shape

- Ensure that icon shapes provide a visual representation matching user expectations and allow association between the icon and function being controlled.

- Design Icon shape as a concrete, not abstract, concept with respect to user.

- Ensure that icon shapes are as simple as possible to ensure user recognition. If icon shape is too complex, the user may make errors in recognizing the icon. Icon shape should show or exaggerate an object's natural features (see Figure 7-8).

- The fewer unique icon shapes used, the more effective the user-computer interface. The most basic (icon) library should be composed of rectangles, triangles, circles, arcs, lines, splines, text, and bitmap images. At a maximum, no more than 20 unique shapes should be used.

- Icons to be used with different cultural or national groups (e.g., NATO Forces), should use technological shapes or forms rather than natural objects. The examples shown in Figure 7-9 include areas shaped as circles, triangles, or boundaries shaped as lines with plus signs (+) as end marks.

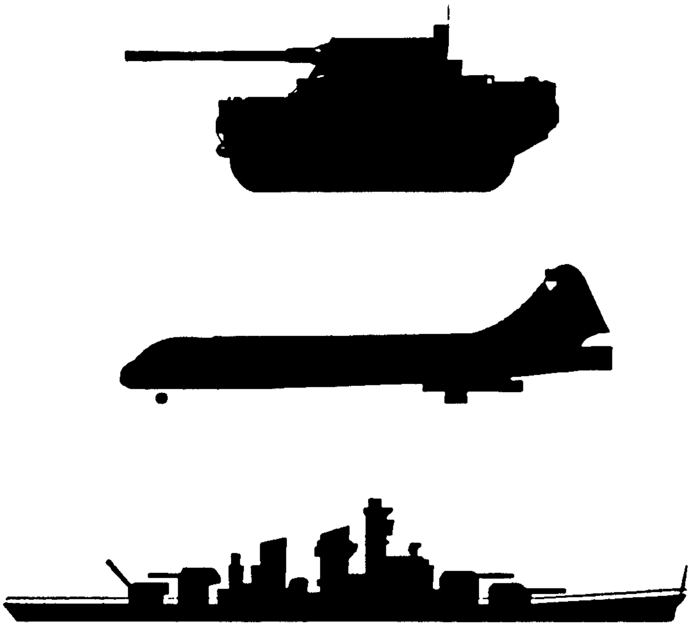


Figure 7-8. Examples of Icon Shapes

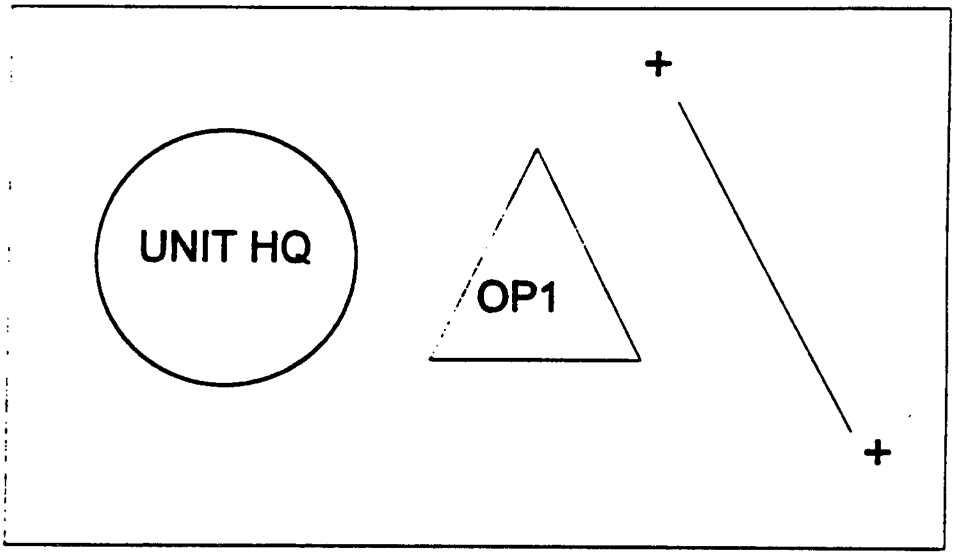


Figure 7-9. Example of the Use of Technological Shapes

- Icons should be consistent with international usage - triangle is "caution" or any pointed object is "enemy," circle with diagonal line through is "prohibit," square is "potential danger or unknown," and round or curved objects are "friendly."

- When icons represent opposite functions, design the icons so they mirror one another (see Figure 7-10).

- Orient figures consistently with respect to text.

- If 3D icons are used, allow the user to manipulate them through rotation so they can be viewed from different vertical and horizontal viewing angles.

7.3.3.5 Icon Size

- Icons should be at least 1/4 inch in height on the screen to reduce the time required for positioning the pointer on the target and performing the required controlling actions.

- When size coding of icons is used by operational systems, the larger icon should be 1.5 times as large as the next smallest.

- Up to five sizes of icon can be used for coding, but no more than three are recommended for operational systems.

- Icons should have a size ratio to the background of 1:1.5 for best visual discrimination.

- Do not use the same symbol in different sizes to mean different things.

- Scales should be kept constant.

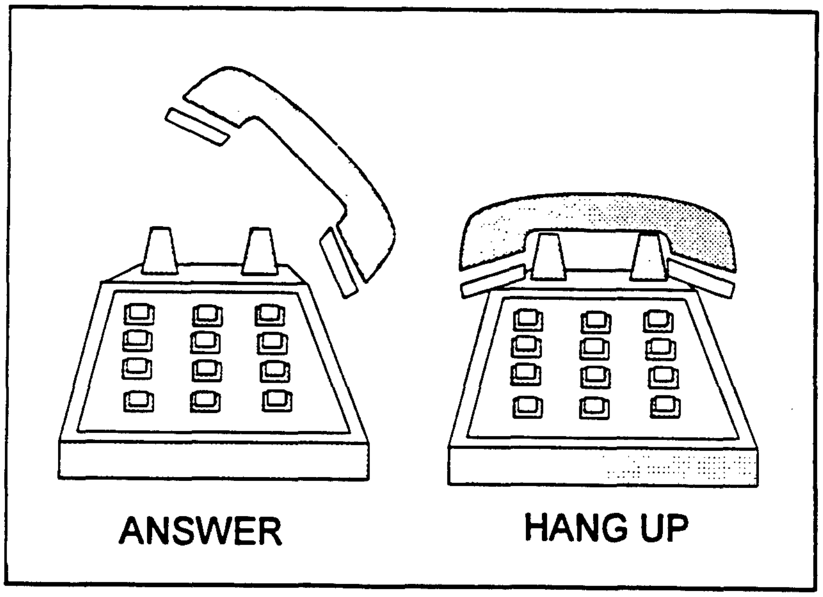


Figure 7-10. Example of Mirrored Icons

7.3.3.6 Grouping Icons

- Base the grouping of icons on the proximity, similarity, and arrangement of objects that define a closed region. Arrange objects in straight or smoothly curving lines. Ensure that symmetrical icon arrangement is maintained when icons undergo simultaneous, correlated changes.

- Grouping can provide additional meaning to icons, but the designer should ensure that unrelated icons are not inadvertently grouped by users.

- Visually separate images in 2-dimensional space.

- Place a finite limit on the number of icons shown at one time.

- Divide the screen into "cells" to improve location and relocation of icons. A well ordered location increases predictability and consistency as well as decreases clutter.

7.3.3.7 Figure-Ground Relationships

Figure refers to an object, which has a shape and stands out from the background. Ground refers to the area that is perceived to continue behind the figure. Keep in mind the following points when designing icons:

- The size of a figure relative to its background is important. The smaller the size of a figure relative to the background, the more likely it will be perceived as a figure.

- When shape only is used as a discriminator for figures, convex shapes will likely be perceived as figures and concave shapes as holes.

- A contour line will be perceived as belonging to only one of the areas it delineates.

- Position affects whether an object will be perceived as a figure. Centrally positioned objects and the lower portion of a surface divided horizontally into two parts are seen as figures.

- The greater the contrast between an object and its background, the greater the perception of the object as a figure.

- Icon figure-ground (foreground lines, etc.) should be clear and stable.

7.3.3.8 Icon Boundary Lines

- Icon boundary lines should be solid and closed and should have a high contrast value (e.g., the best boundary is based on the contrast between the figure and the underlying display background).

- Use an optically consistent line weight for unity; use curves to enhance visual relationships.

- Corners should be smooth.

- It is best not to put a box around an icon. The box makes visual discrimination more difficult.

- If a boundary is left open, be aware that the user will tend perceptually to close the open boundary. Design the opening to ensure only desired closures will occur.

7.3.3.9 Meaning

The user associates a meaning with an icon. Icons can be a very powerful form of communication since they have the potential of being universally meaningful.

- The meaning should be inherently obvious. The stronger the associated meaning, the more easily the icon will be recognized and remembered.

- Icons should have intrinsic meaning to the user. Care must be taken to ensure no negative connotations can be attributed to the icon.

- Avoid icon designs that could have a range of attributed meanings. The user should not have to look up a meaning or activate an icon to understand what it does or means.

- Carefully consider icon style. Representational and abstract symbols have visual resemblance; arbitrary or invented symbols must be learned.

- Learned icons should be as unique or compact as possible to aid in training the user.

- Icons should not be too realistic, stylized, simple, or complex.

- Design icons as concretely as possible. Subjects tend to rate less concrete icons as more appropriate when they have been redesigned to be more concrete. Appropriateness has been found to be a reasonable predictor of icon identification time.

- Grouping should be used to provide additional meaning to icons.

7.3.3.10 Hardware Considerations

- Computer monitors should offer sufficient resolution so that the icons can be identified by the user at normal viewing distance.

- On high-resolution screens, at 60-150 dots per inch (dpi), 30-60 pixels per inch are often used.

- The designer should keep in mind that display standards, such as color graphics adapter (CGA), enhanced graphics adapter (EGA), and video graphics array (VGA) screen resolutions, are all different; some even have differently shaped pixels. Icons must be designed to appear correctly in each of these screen display standards.

- If icon display equipment has severe limitations in appearance or interaction characteristics (e.g., monochrome CRTs or touch-screen input), this will affect the appearance of icons and their use by the viewer.
- The designer should consider the intended display medium when designing the icon (e.g., an icon design for a high-resolution display may lose meaning when presented on a low-resolution display).

7.3.4 Design Methodology

Successful icon design involves approaching the problem systematically by analyzing the sometimes conflicting needs that determine appearance and interaction characteristics, designing prototypes, and evaluating the design.

Although no set of rules can guarantee a perfectly designed icon or set of icons, the following general design steps are suggested.

7.3.4.1 Analyze Contents

Analyze the verbal contents and the display environment to determine how icon parts and complete icons should relate.

7.3.4.2 Use Sketches

Specify appearance of the icon, placing and shaping instances of existing icons. Design the initial icons by creating quick sketches showing all visual elements, their approximate size, and approximate location. It is easier to manipulate broad differences in icons and their hierarchy early in the design process, but avoid becoming too precise. Explore all possible design variations.

7.3.4.3 Establish Style

A consistent stylistic treatment has a perceived complexity of the icons. Styles should be established in which the icons are grouped by consistent approach or appearance. User involvement is a critical factor at this point.

7.3.4.4 Establish Layout

Consider the following issues for screen layout design:

- An underlying grid helps organize major elements of the icons to make all visual components consistent (e.g., point elements, gray patterns, curves, angles, length and width of rules).

- Establish standard horizontal, vertical, and oblique lines and a limited set of sizes for objects. Use the grid to regulate groups of text and images and determine the size of elements in order to build a visual consistency.

- Use an articulate, systematic method of assigning areas for text and illustration, as well as for background field or format. When possible, use strong, easily recognized proportional format.

- Research concerning the user's ability to select images on a CRT screen supports the interpretation that the location of icons is of high importance; frequently used icons should probably be positioned around/near the edges of the screen.

- It is important to make room for the most important icons in the same places on screen (e.g., the trash bin).

7.3.4.5 Distinguish Icons

Consider the following items to distinguish icons.

- Use large objects, bold lines, and simple areas.

- Select a style of presentation, and continue to use it within the icon set.

- Avoid sudden changes in emphasis or de-emphasis of certain objects, structures, or processes.

- Ensure crucial elements are of sufficient size in comparison to the total size of the icon.

7.3.5 Icon Evaluation

The following are some issues to use for evaluating symbols:

- Is symbol/message association easy? Representational and abstract symbols have visual resemblance; arbitrary or invented symbols must be learned (e.g., math symbols). Arbitrary symbols should be as unique or compact as possible, as they require training.

- In a variety of cultures and situations, is the symbol equally appropriate? Icons created for one cultural group may generate incongruent or even opposite meaning for another group. Such differences may generate conflicts in communications.

- Will the symbol be appropriate in the future? Will the metaphor soon be obsolete?

- Is the symbol pleasing and noncontroversial?

- Is the symbol in accordance with existing international standard symbols (i.e., do not create new symbols if one already exists)?

- Can the symbol or its elements be applied systematically for a variety of interrelated concepts (i.e., can the elements form a rich symbolic language, combining them to form more complex symbols)?

- Is the symbol easy to reproduce in a variety of environment and situations? Can it be transferred to different systems, enlarged or reduced without losing crispness or detail?

- Is the symbol distinguishable from other symbols?

- Can the symbol be perceived from different distances, angles, conditions?

- Do icons have intrinsic meaning to the user? The user will associate a meaning with an icon. The stronger the associated meaning, the more easily the icon will be recognized and remembered.

- Do icons provide a visual representation that matches user expectations and allows association between the icon and the function being controlled?

7.3.5.1 Testing Icon Design

It is imperative to test icon design with a group of users who represent the intended user. This should ensure that the meaning of the icon is implicitly understood.

7.3.5.2 Usability

- Icons should be general enough to allow the user to understand and use other metaphors or media, such as text-based systems.

- Application software should allow the creation of icons that represent macro instructions. The user will be able to use these macros more effectively with the advantages of a visual representation.

REFERENCES

Paragraph	References
7.0	Helander (1988); Gittins (1986); Lewis and Fallesen (1989); DISA/CIM (1992c) pp. 7-1 to 7-3
7.1	Smith and Mosier (1986) para 3.1.8-5
7.1.1a	Ziegler and Fähnrich (1988) p. 127
7.1.1b	Ziegler and Fähnrich (1988) p. 127
7.1.1c	Ziegler and Fähnrich (1988) p. 128
7.1.2	Ziegler and Fähnrich (1988) p. 128
7.1.3	Ziegler and Fähnrich (1988) p. 129
7.1.4	Bowser (1991)
7.2	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 519; Lerner and Collins (1980) p. 32; Tzeng et al. (1990) pp. 77-97
7.2.1	Arnstein (1983) preface
7.2.1.1	OSF (1990) p. 531
7.2.1.2	Lewis and Fallesen (1989) p. 92
7.2.1.3	DISA/CIM (1992c) pp. 7-2
7.2.1.4	DISA/CIM (1992c) pp. 7-2
7.2.1.5	Lewis and Fallesen (1989) p. 92; Gittins (1986) p. 530 and 540
7.2.2.1	Gittins (1986) p. 528
7.2.2.2	OSF (1990) p. 531
7.2.2.3	Lewis and Fallesen (1989) p. 91
7.2.2.4	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 540
7.2.2.5	OSF (1990) p. 530
7.2.2.6	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 540
7.3	AIGA (1982) preface, p. 129; Marcus (1979) p. 26; Tzeng et al. (1990) p. 78; Lerner and Collins (1980) p. 32
7.3.1	AIGA (1982) p. 11; Wood and Wood (1987) p. 100; Rogers (1989) p. 110

REFERENCES (cont'd)

Paragraph	References
7.3.2	Ziegler and Fähnrich (1988) p. 127; Jorna (1988) p. 182
7.3.2.1a	Smith and Mosier (1986) para 3.1.8-5; Wood and Wood (1987) p. 100
7.3.2.1b	Gittins (1986) p. 532
7.3.2.1c	Gittins (1986) p. 532
7.3.2.1d	Gittins (1986) p. 535; Blankenberger and Hahn (1991) p. 376
7.3.2 1e	Bullinger et al. (1987) p. 90; Gittins (1986) p. 536
7.3.2.1f	DISA/CIM (1992c)
7.3.2.1g	Gittins (1986) p. 526
7.3.2.1h	Lewis and Fallesen (1989)
7.3.2.2	DISA/CIM (1992c)
7.3.2.2a	DISA/CIM (1992c)
7.3.2.2b	Wood and Wood (1987) p. 103
7.3.2.2c	DISA/CIM (1992c)
7.3.2.2d	DISA/CIM (1992c)
7.3.2.3	WordPerfect Corporation (1991) pp. 35-42
7.3.2.4	Rogers (1989) p. 110
7.3.2.5	Ziegler and Fähnrich (1988) p. 128
7.3.3	Marcus (1992) p. 35; Nolan (1989) p. 381; Lewis and Fallesen (1989)
7.3.3.1a	Lewis and Fallesen (1989) p. 89; MacGregor, King, and Clarke (1988) p. 275
7.3.3.1b	Lewis and Fallesen (1989) p. 89
7.3.3.1c	Wood and Wood (1997) p. 101
7.3.3.1d	Wood and Wood (1997) p. 101
7.3.3.1e	Neurath (1980) p. 49
7.3.3.1f	Stammers and Hoffman (1991) pp. 354-356
7.3.3.1g	Rogers (1989) p. 106

REFERENCES (cont'd)

Paragraph	References
7.3.3.1h	Stammers and Hoffman (1991) pp. 354-356
7.3.3.1i	Gittins (1986) p. 526
7.3.3.1j	Rosenstein and Weitzman (1990) p. 525
7.3.3.1k	DISA/CIM (1992c)
7.3.3.1l	DISA/CIM (1992c)
7.3.3.1m	Wood and Wood (1987) p. 103
7.3.3.2a	Tullis (1981) p. 548
7.3.3.2b	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 539
7.3.3.2c	Marcus (1992) p. 63
7.3.3.2d	Marcus (1992) p. 63
7.3.3.2e	Marcus (1992) p. 63
7.3.3.2f	Marcus (1984) p. 26
7.3.3.3a	Lewis and Fallesen (1989); Ziegler and Fähnrich (1988) p. 128; Smith and Magee (1980) p. 384
7.3.3.3b	Lewis and Fallesen (1989) p. 91
7.3.3.3c	Neurath (1980) p. 24
7.3.3.4a	Ziegler and Fähnrich (1988) p. 128; Smith and Mosier (1986) para 2.4-13; Shneiderman (1987) p. 200; Lewis and Fallesen (1989) p. 89; Lansdale (1988) p. 148
7.3.3.4b	DISA/CIM (1992c)
7.3.3.4c	Ziegler and Fähnrich (1988) p. 128; Shneiderman (1987) p. 200; Lewis and Fallesen (1989) p. 89; Gittins (1986) p. 528
7.3.3.4d	Lewis and Fallesen (1989) p. 91; Rosenstein and Weitzman (1990) p. 525
7.3.3.4e	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 539
7.3.3.4f	Wiebe et al. (1993) p. 55
7.3.3.4g	Lewis and Fallesen (1989) p. 90
7.3.3.4h	Marcus (1979) p. 28

REFERENCES (cont'd)

Paragraph	References
7.3.3.4i	Lewis and Fallesen (1989) p. 92
7.3.3.5a	Lewis and Fallesen (1989) p. 90
7.3.3.5b	Lewis and Fallesen (1989) p. 90
7.3.3.5c	Lewis and Fallesen (1989) p. 90
7.3.3.5d	Lewis and Fallesen (1989) p. 90
7.3.3.5e	Neurath (1980) p. 47
7.3.3.5f	Marcus (1979) p. 26
7.3.3.6a	Lewis and Fallesen (1989)
7.3.3.6b	Lewis and Fallesen (1989)
7.3.3.6c	Snyder (1987) p. 147
7.3.3.6d	Gittins (1986) p. 532
7.3.3.6e	Gittins (1986) p. 532
7.3.3.7	Lewis and Fallesen (1989); Gittins (1986) p. 539
7.3.3.8a	Lewis and Fallesen (1989) p. 90; Gittins (1986) p. 539
7.3.3.8b	AIGA (1982) p. 129
7.3.3.8c	Lewis and Fallesen (1989) p. 90; Gittins (1986) p. 539
7.3.3.8d	Nolan (1989) p. 381
7.3.3.8e	Gittins (1986) p. 539
7.3.3.9	Lewis and Fallesen (1989); Rogers (1989) p. 106
7.3.3.9a	Lewis and Fallesen (1989)
7.3.3.9b	Lewis and Fallesen (1989); Lodding (1983) p. 19
7.3.3.9c	Rogers (1989) p. 106
7.3.3.9d	Wood and Wood (1987) p. 100
7.3.3.9e	Wood and Wood (1987) p. 100
7.3.3.9f	Lodding (1983) p. 14
7.3.3.9g	Stammers and Hoffman (1991) p. 354
7.3.3.9h	Lewis and Fallesen (1989)

REFERENCES (cont'd)

Paragraph	References
7.3.3.10a	Marcus (1992) p. 55
7.3.3.10b	Marcus (1992) p. 55-56
7.3.3.10c	Marcus (1992) p. 56
7.3.3.10d	Marcus (1992) p. 61
7.3.3.10e	Ziegler and Fähnrich (1988) p. 127
7.3.4	Marcus (1992)
7.3.4.1	Marcus (1992)
7.3.4.2	Marcus (1992)
7.3.4.3	Marcus (1992)
7.3.4.4	Marcus (1992)
7.3.4.4a	Marcus (1992); Marcus (1984) p. 26
7.3.4.4b	Marcus (1992); Marcus (1984) p. 26
7.3.4.4c	Marcus (1992)
7.3.4.4d	Blankenberger and Hahn (1991) p. 367
7.3.4.4e	Blankenberger and Hahn (1991) p. 374
7.3.4.5	Marcus (1992)
7.3.4.5a	Marcus (1992)
7.3.4.5b	Marcus (1992); Jorna (1988) pp. 173-183
7.3.4.5c	Marcus (1992)
7.3.4.5d	Marcus (1992)
7.3.5	Marcus (1992); Kato (1972) p. 100
7.3.5a	Marcus (1992)
7.3.5.b	Marcus (1992)
7.3.5.c	Marcus (1992)
7.3.5.d	Marcus (1992)
7.3.5.e	Marcus (1992)
7.3.5.f	Marcus (1992)

REFERENCES (cont'd)

Paragraph	References
7.3.5g	Marcus (1992)
7.3.5h	Marcus (1992)
7.3.5i	Marcus (1992)
7.3.5j	Marcus (1992); Lewis and Fallesen (1989)
7.3.5k	DISA/CIM (1992) p. 7-14
7.3.5.1	Smith and Mosier (1986) para 2.4-13; Lerner and Collins (1980) p. 32
7.3.5.2a	DISA/CIM (1992) p. 7-14
7.3.5.2b	DISA/CIM (1992) p. 7-14

8.0 COMMON FEATURES

This section describes features, functions, and field display formats that should be handled consistently by all DoD applications. Subsection 8.1 deals with issues that apply primarily to operational systems. Subsections 8.2, 8.3, and 8.4 apply to all DoD systems. Subsection 8.2 discusses the topic of HELP. Subsection 8.3 discusses those characteristics of interactive control that apply to all dialog types. Subsection 8.4 discusses function keys. Additional information on interactive dialog can be found in documents such as Smith and Mosier (1986), Helander (1988), or DoD (1989b).

8.1 TACTICAL SYSTEM COMMON FEATURES

This section describes features, functions, and field display formats that should be handled consistently by all DoD operational applications.

8.1.1 Date/Time Display

When date and time information are displayed in digital form, the format should be as follows:

8.1.1.1 Date

Use YYYYMMDD, where YYYY is the four digits of the year, MM is the month, and DD is the day (e.g., 19910104 specifies 4 January 1991), or DD MMM YYYY, where DD is the day, MMM is the month, and YYYY is the year (e.g., 04 JAN 1991). With the year 2000 approaching, it is widely recognized that storing only two digits to denote a year will cause serious system problems. The display and data fields of a date should comply with a four-digit entry for the year.

8.1.1.2 Time

Use HHMM{SS}Z, where HH is the hour of a 24-hour day, MM is the minute, SS (optional) is the second, and Z is the time zone. Zulu (Z), or Greenwich Mean time in civilian terms, is the system standard and the default DoD display standard (e.g., 113024Z). Unless otherwise specified, use colons or spaces on the display or output format to make the format more readable (e.g., 11:30:24Z). To simplify data entry and avoid extraneous characters, generate colons or spaces as part of the form, and do not leave this to user discretion.

8.1.1.3 Local Time

Allow users to specify local time on hard-copy output and soft-copy display, as desired (e.g., 11:30:24L). However, do not provide this option to users in operational systems where input and coordination are based on Zulu time.

8.1.1.4 Date/Time Group

Military services specify that Date/Time Group should be displayed as DDHHMMZ MMM YY, where DD is the day, HH is the hour of a 24-hour day, MM is the minute, Z is the time zone (defaults to Zulu), MMM is the month, and YY is the year (e.g., 041130Z JAN 91). This format for the display of a date in no way implies that the data field should use a two-digit field for the year. While displays of the year may be abbreviated, data fields need to comply with a four-digit entry for the year.

8.1.2 Latitude/Longitude Display

Latitude and longitude displays will always be presented as two fields. The labels may be given as Lat and Long. When displaying latitude and longitude, use appropriate symbols for degrees, minutes, and seconds as part of the display. The formats are shown in the following two sections.

8.1.2.1 Latitude

Use D{D}H, where D (one or two characters) is the degrees of latitude and H is the hemisphere (N for North, S for South), or DD(MM{SS})H, where DD is the degrees of latitude, MM is the minutes of latitude, SS is the seconds of latitude, and H is the hemisphere (N for North, S for South).

8.1.2.2 Longitude

Use D(D{D})H where D (one, two, or three characters) is the degrees of longitude and H is the hemisphere (E for East, W for West), or DDD(MM{SS})H where DDD is the degrees of longitude, MM is the minutes of longitude, SS is the seconds of longitude, and H is the hemisphere (E for East, W for West).

8.1.3 User-Definable Parameters

Enable all users to configure their computer screens to individual preferences. User-definable parameters include, but are not limited to, those that follow.

8.1.3.1 Display Colors

Where feasible, users should be able to select map and window background colors from a color palette within a user-parameter selection window. The selected color should be immediately reflected in a sample item displayed within the selection window. However, users should not be allowed to change security banner colors or colors with specific tactical coded meaning. Other restrictions are noted in the service-specific addenda to this *Style Guide*.

8.1.3.2 Printer Default

In networked environments, enable users to specify the printer destination.

8.1.3.3 Mouse Button Function Mappings

Enable users to specify either left-handed or right-handed button configurations as defined in Section 3.0 of the *Style Guide*.

8.1.3.4 HELP Level

Enable experienced users to bypass novice-level HELP messages that are beneficial to a new user.

8.1.4 Wild-Card Characters

Use wild-card characters in queries and searches to support patterns. The use of wild cards is application-specific. Some applications may disallow wild cards or restrict their use to only a few of the following wild card conventions. Use the following conventions where possible.

8.1.4.1 Single Alphabetic Character

Use an @ to replace any single alphabetic character (a-z and A-Z). For example, an input of abc@d would match abcad, abced, and abczd, but would not match abc7d or abcddd.

8.1.4.2 Single Numeric Character

Use a # to replace any single numeric character (0-9). For example, an input of 123#4 would match 12334, 12394, but would not match 123x4 or 123554.

8.1.4.3 Single Alphanumeric Character

Use a ? to replace any single alphanumeric character (a-z, A-Z, 0-9, and punctuation marks). For example, an input of abc?d would match the character strings abcad, abc(d, abc'd, and abc7d, but would not match abcxxxd.

8.1.4.4 String

Use an * to replace zero or more alphanumeric characters. For example, an input of abc*d would match the character strings abcad, abcd, abckjfi(rjk)fid, and abc7d, but would not match abcd5.

8.2 ON-LINE HELP

On-line help (HELP) provides procedural aids, the ability to recover from errors, and advice without requiring the user to exit the application. Ideally, HELP is always available. A well designed system offers context-sensitive HELP.

Two elements are critical to HELP: user interface and content; both are equally important (Kearsley 1988). HELP must be easy to use and provide readily understandable user guidance;

HELP must not add problems or make a user situation more confusing. The HELP interface design will contribute to how often HELP is used, because the more difficult the interface is to use or access, the higher the probability HELP will not be used. No HELP application is useful if difficult to obtain, hard to use, or difficult to return to the application program.

Computer users want to accomplish a particular task quickly and with the least effort possible. When users encounter a problem, they want a solution that involves minimal interruption of the task at hand. If information is not immediately available, users often guess, repeat a previous sequence, or ignore what is not understood. These responses usually lead to further problems.

Along with individual differences, users have varying degrees of computer experience. The following three types of user group may require different types or levels of HELP:

- Novices (users who have little experience with computers) need help with basic concepts and operations. Novices usually want to see only necessary information.

- Experts (experienced computer users) want to know about limitations, shortcuts, complex operations, and anything else that will allow them to do their work more efficiently.

- Casual users (who may be either novices or experts) only occasionally use a computer or current application. They may need help remembering aspects of the application they previously learned.

General guidelines:

- Make HELP easy for users to access.

- Make HELP available throughout the application.

- Make access to HELP uniform.

- Make HELP easy to understand.

- Make it easy to return to the application.

8.2.1 Types Of HELP

HELP should reflect user requirements with no significant impact on application response time. Of the following three types of HELP, the advice and active forms are preferred. Embedded training is often included in HELP. However, it is recommended that embedded training not be combined with HELP.

8.2.1.1 Advice

Enable users to obtain advice from HELP. As users query HELP, they find an interactive, context-sensitive source of information that indicates the entry to make at the current location in the application, the required keystroke, or the steps to take to complete the task.

8.2.1.2 Active

Ensure that HELP is active, such that when HELP application software senses an inappropriate entry, it interrupts to ask users what they are attempting and if they are sure they want to complete the operation they initiated. HELP then suggests the correct form or keystroke.

8.2.1.3 Passive

Enable users to query HELP when they need assistance. The information may be in the form of on-line system documentation, such as a user's guide or a list of functions performed by combinations of keypresses.

8.2.2 General Design

8.2.2.1 Minimize Keystrokes

Provide single keystroke access to and exit from HELP.

8.2.2.2 Provide Memory Aids

Assume users cannot remember everything required to run the application; provide memory aids.

8.2.2.3 Include Basic Information

Include basic information you would expect only novices to seek.

8.2.2.4 Expand Upon the Manual

Provide clearer explanations of information in the manual, using subsequent screens as needed. Do not simply repeat phrases from the manual that the user has read but may not understand.

8.2.2.5 Choose On-line Portions of the Manual Selectively

Be selective when putting information on-line from the user manual. Do not put the entire manual on-line as this would make it more difficult to navigate and read through than the hard-copy version. It would also waste system memory.

8.2.2.6 Include Obvious Information

Include all pertinent information, even that which may appear obvious to the developer.

8.2.2.7 Avoid Jargon

Avoid using jargon. A friendly, effective interface is the most important component of a HELP system. It frustrates a naive computer user to type "HELP," then receive a bit of cryptic jargon in reply. Use jargon common to all users, not of the designer or programmer, when use of jargon is unavoidable.

8.2.2.8 Do Not Overload the User

Do not expect the user to read more than three HELP displays at a time or to remember more than about five points.

8.2.2.9 Do Not Use HELP to Teach

Do not use HELP to teach novices how to operate the system. Provide step-by-step instructions to remind occasional users how to perform the most common tasks. Remember that most users perform the same few tasks over and over, in the simplest possible way.

8.2.3 Accessibility Of HELP

8.2.3.1 Universal Access

Provide access to HELP from every screen.

8.2.3.2 Availability

Remind users that HELP is easily available by displaying the command or function key used to get HELP.

8.2.3.3 Display HELP Status

Display a message indicating the status of HELP availability, if HELP is not available at all times or places in the program.

8.2.3.4 Single Action to Invoke

Enable users to get HELP using only a single keypress or mouse-click.

8.2.4 Provide HELP on HELP

8.2.4.1 Alphabetical Index of Functions

Make an alphabetical index of HELP functions available to the user.

8.2.4.2 Alphabetical Index of Commands

Provide an alphabetical index with explanations of all commands used by the application software, showing the argument options.

8.2.4.3 Show How to Use

Show users how to use the HELP function. Never assume that HELP is obvious, even to expert users.

8.2.4.4 Present Alternatives

Show how to get HELP from anywhere in the system. Because users may know only one route, detail alternatives, including how quick and easy it is to use the options. Define different meanings of the HELP display, and explain their functions.

8.2.4.5 Navigating Through HELP

Show how to navigate within HELP. Explain how to scroll or page through a topic and how to jump to related topics.

8.2.4.6 Provide HELP on Screens and Windows

Describe the current window, including its function and tasks the user can perform.

8.2.4.7 Provide Instructions to Novices

Place instructions for using HELP on every HELP display, to assist novice and casual users.

8.2.4.8 Instruct on When to Use

Provide users with complete instructions on when to use the information supplied by HELP.

8.2.5 Application Information

Provide a list of application capabilities. Show application components, options, and structure to help the user understand the application and use it more effectively. Experienced users as well as novices underutilize many applications because they do not recognize the full range of capabilities.

8.2.5.1 Provide Shortcuts

Use HELP to point out shortcuts and unused features.

8.2.5.2 HELP on Messages

Make available successively more detailed explanations of a displayed error message. HELP should be considered to provide more detailed messages, such as information and status.

8.2.5.3 HELP on Prompts and Definitions

Make available successively more detailed explanations of a displayed question or prompt and definitions of specified terms.

8.2.5.4 Show Correct Input

Provide examples of correct input or valid commands.

8.2.5.5 Show Command Format

Provide a description of the format of a specified command and a list of allowable commands.

8.2.5.6 Provide User-Centered HELP

Ensure that HELP is user-centered; base HELP on the user's task, not on application characteristics. Descriptions of application characteristics are more appropriate for a hard-copy user's manual.

8.2.6 Provide HELP In Context

Context-sensitive HELP may be the most important kind of HELP for users. Ensure that context-sensitive HELP describes the nature of a specific control (check button, radio button, slider bar) and how people use that control.

8.2.6.1 Provide Specific HELP

Ensure the HELP is specific to each level of user interaction (e.g., for context-sensitive HELP in a field specifying printer baud rate).

> *Note: "Baud rate is the speed in bits per second at which your printer can accept data. Acceptable speeds are 1200, 2400, and 9600. Enter the speed in bits per second."*

8.2.6.2 Show Correct Alternatives

List correct alternatives if the user enters an incorrect command.

8.2.6.3 Provide HELP Within Application

Provide HELP within the application so users do not have to abandon their place in the application to seek HELP. Ensure that users do not have to close files, exit the application, and/or log off to invoke a HELP utility.

8.2.6.4 Use Split Screen or Window

Allow users to see the application screen that relates to the HELP request by means of a split screen or window. A separate HELP screen that completely replaces the application screen is undesirable because it prevents the user from simultaneously observing the problem and the HELP screen.

8.2.6.5 Resize and Reposition HELP Windows

Provide the user the capability to resize and reposition windows to see the HELP information and the problem at the same time.

8.2.6.6 Identify Special Keys

Display the meanings assigned by the application where applications have special uses for keys, especially function keys.

8.2.7 User Control of the HELP System

Give users more control over a HELP system, as they will find this more useful.

8.2.7.1 User-Initiated

Allow users to initiate a HELP request and select the desired HELP topic.

8.2.7.2 User-Selected Levels

Allow users to select a level of HELP if multiple levels are available.

8.2.7.3 Annotate Messages

Allow users to annotate existing HELP messages.

8.2.7.4 Describe Key Functions

Provide the capability within HELP of pressing any key to obtain a list of features whose names begin with that letter. When the user selects a feature from the list by highlighting or clicking, provide an explanation of the feature.

8.2.8 Provide Consistent HELP Format

8.2.8.1 Consistent Screens

Provide consistent HELP aids from screen to screen, both with indicators that HELP is available (e.g., "F1 = HELP") and the specific location on the screen.

8.2.8.2 Progressive Detail

When providing progressively more detailed explanations, ensure the process of moving from level to level is consistent from screen to screen.

8.2.9 Self-Explanatory and Concise Displays

8.2.9.1 Match Titles to Contents

Reflect or match the content of a HELP window in its title (e.g., the title of a HELP window for the entry field "Trans" could be "Help for Trans").

8.2.9.2 Match Names

Ensure that the name on the HELP display matches the panel from which help was requested (e.g., when working on an accident report, the help display may read, "HELP: ACCIDENT REPORT").

8.2.9.3 Ensure Relevancy to the User

Tailor the display to the current information requirements of the user, so only relevant data are displayed.

8.2.9.4 Provide Clear Messages

Make error and HELP messages clear, concise, and appropriate to the experience and training users have had in using the system.

8.2.9.5 Use Task-Oriented Wording

Adopt task-oriented wording for labels, prompts, and user guidance messages, incorporating whatever special terms and technical jargon may be normally employed in the user's tasks.

8.2.9.6 Increase Understandability

To increase understandability of HELP, apply the following principles:

- Use short sentences when writing HELP messages.
- Use the active voice in all HELP messages.
- Provide as many examples as possible for each HELP screen.
- Place HELP information in tables, where applicable.
- Put the answer before the explanation when presenting HELP information.

- Answer the most likely HELP questions immediately.
- Minimize the user requirement to scroll or page through displays.

8.2.10 Make Return To Application Easy

8.2.10.1 Single Keystroke

Enable the user to return to the application with only a single keypress or mouse-click.

8.2.10.2 Exit HELP Easily

When a single keystroke exit is not possible, enable the user to return to the application easily. Ideally, this would be accomplished without calling up a menu and then choosing an item from it.

8.2.11 Keep HELP Current

8.2.11.1 Provide Up-to-Date HELP

Plan and build HELP concurrently with developing applications, so HELP information reflects the current version of the software. Provide updates to HELP with subsequent software releases.

8.2.11.2 Tailor HELP to the User

Collect data on user target population to tailor HELP to the training and experience of the users.

8.2.12 Provide User Options

8.2.12.1 Bookmarking

Provide "bookmarking" so users can flag specific HELP messages for easy referral later. This can be especially useful in a large help system consisting of many topics and screens. Bookmarking allows users to customize the HELP system to their own needs and filter out information of no interest; it can speed up the HELP process and return the user to work with fewer interruptions.

- Allow the user to select a bookmark option while viewing the message to flag a HELP message.
- Ensure that user has an option to see all or just the bookmarked messages.
- Ensure a print option is available while HELP messages are being displayed. Users often want to print out HELP information to study it further.

8.2.13 System-Initiated Messages

Provide system-initiated messages when an error has been detected or when other evidence reveals that the user is having a problem (e.g., missing parameters, duplicating erroneous commands, long lapses in response, out-of-range responses).

8.2.13.1 Positive Tone

Ensure that messages have a positive tone, indicating what must be corrected. Focus on correction(s) to the problem, not the action that caused it.

8.2.13.2 User Control

Present system-initiated messages to users as advice or suggestions. Ensure that HELP messages are not intrusive.

8.2.13.3 Error Control

Provide system-initiated HELP messages for systems where incorrect user actions could result in serious consequences. This is especially true for destructive actions such as deletions (e.g., MS-DOS command: "Del *.*"), file replacements, exiting an application without saving data, or renaming a file.

8.2.13.4 Document Errors

In error messages, always state the error detected, the input field containing the error, and the corrective action.

8.2.13.5 User Understanding of Message

Indicate clearly in messages whether they are meant to inform of error, indicate status, prompt for action, or provide feedback.

8.2.13.6 User Options

Give users the option to turn off system-generated messages or to specify the level or type of message to be given (e.g., advisory, caution, warning).

8.2.13.7 Avoid Jargon

Ensure that error messages are specific and address the problem in user terms. Avoid vague terms, such as "syntax error," or obscure error code numbers.

8.3 INTERACTIVE CONTROL

Interaction between the computer and the user is performed through a two-way communication process: 1) the user inputs commands, and 2) the computer responds to the input. Generally, two interchangeable names are given to this process -- sequence control (Smith and Mosier 1986) and interactive control (DoD 1989a). The term "interactive control" will be used in this *Style Guide*.

- Interactive control of a system occurs through a give-and-take of command and response between the user and the computer, called a "dialog." The following have been identified as the eight major types of user-computer dialogs (Smith and Mosier 1986):

  - **Question and Answer** - The user responds to questions posed by the computer.
  - **Form Filling** - The user enters a series of commands or data items in predefined fields. These fields may be mandatory or optional.
  - **Menu Selection** - The user selects from predefined option lists by pointing with a device, such as a mouse, or keying in associated codes.
  - **Function Keys** - The user controls the dialog by using fixed or variable function keys on the keyboard.
  - **Command Language** - The user makes control entries by composing specified messages for the computer.
  - **Query Language** - The user employs a specialized type of command language to elicit information from a computer system. This is used extensively with databases.
  - **Natural Language** - The user can compose messages to control the computer based on natural, not specialized, languages.
  - **Graphical Interaction** - The user makes selections and controls the computer interaction by direct manipulation.

Each of these eight types of dialogs can be used individually or combined as a suite or set of techniques. OSF/Motif, for example, supports a combination of menu and graphical interaction techniques.

Eight principles form the basis for designing a good human-computer dialog (Shneiderman 1987; Bailey 1982). These principles are as follows:

- Strive for consistency of design across terminology, menus, command structure, etc. for all applications.
- Enable frequent users to use shortcuts, improving user acceptance and overall system performance.
- Offer informative feedback for all user actions.

- Design dialogs to yield closure. The user will then feel a sense of accomplishment and will know when to go on to the next task.

- Offer simple error-handling, both by system error-checking and ease in correcting an identified error.

- Allow easy reversal of actions, such as an UNDO capability.

- Enable the user to feel in control of the interaction with the system.

- Reduce short-term memory load on the user by using intuitive displays, interactive sequences, sufficient training, and on-line helps and tutorials.

The primary dialog types used by applications share certain design considerations and guidelines. These are addressed in this section and organized into six topics: general, context definition, transaction selection, interrupts, error management, and alarms.

8.3.1 General

The following general guidelines for interactive control apply to DoD systems.

8.3.1.1 Displayed Context

If the results of a control entry vary depending on a prior action of the user or computer, display a continuous indication of the current context (mode).

8.3.1.2 Irrelevant Data

Provide the user with the capability to remove irrelevant items from the display and to reverse this action (i.e., retrieve information that was removed).

8.3.1.3 Page-Back Capability

When the requested data exceed the capacity of a single display frame, provide the user with easy methods to move back and forth over displayed material by paging or panning/scrolling.

8.3.1.4 Upper and Lower Case Equivalent

For interpreting user-composed control entries, treat upper and lower case letters as equivalent.

8.3.1.5 User-Callable Unfamiliar Term Descriptions

- Write interface dialog to provide the capability for the user to call up descriptions of unfamiliar terms and commands through context-sensitive HELP screens. See Subsection 8.2 for additional information.

- Ensure that the interface displays to the user, as needed and in immediately usable form, terminology and commands necessary to perform the task associated with the displayed information.

8.3.1.6 User-Paced Sequence Control

Allow users to pace control entries, rather than forcing them to keep pace with computer processing or external events.

8.3.1.7 Logical Transaction Sequences

Design the sequence of transactions (e.g., number and sequence of steps in a task) from the perspective of what is logical to the user, not what is logical from the perspective of computer processing or ease of programming.

8.3.1.8 Automated Information Entry

Routinely and automatically include informational elements required for every communication or transaction after first input (e.g., call signs and authentication procedures).

8.3.1.9 Customized Display/Control Options

Allow the user to customize the information displayed on a screen to the particular tactical mission or scenario. For example, the user should have the flexibility to define which files can be displayed concurrently and what tactical data will be utilized in a single display. See Figure 8-1 for examples.

8.3.1.10 Distinctive Display of Control Information

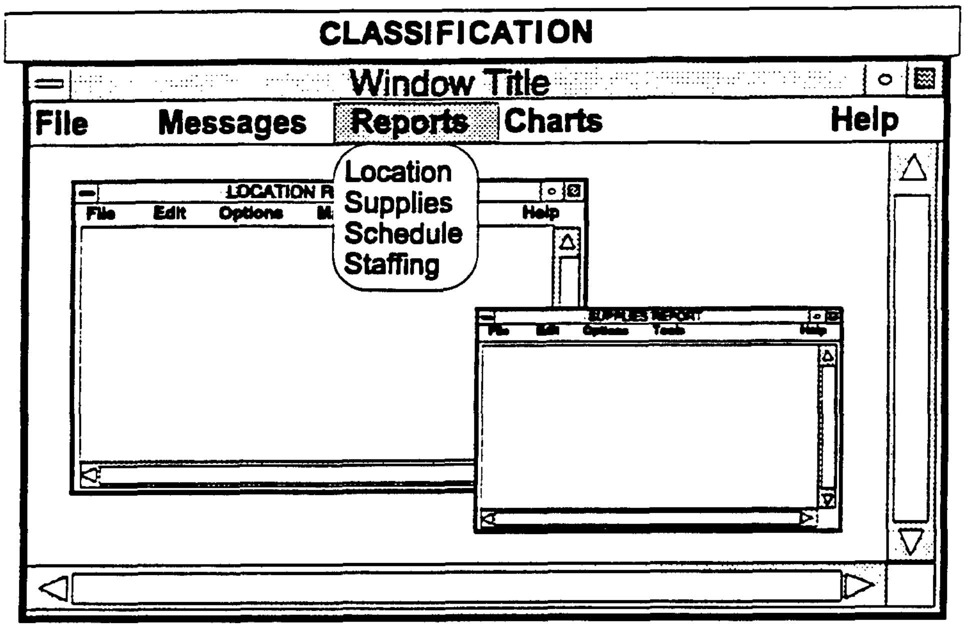
Design all displays so features (e.g., prompts, advisories, etc.) relevant to interactive control are distinctive in position and/or format.

8.3.1.11 System Matched to User Abilities

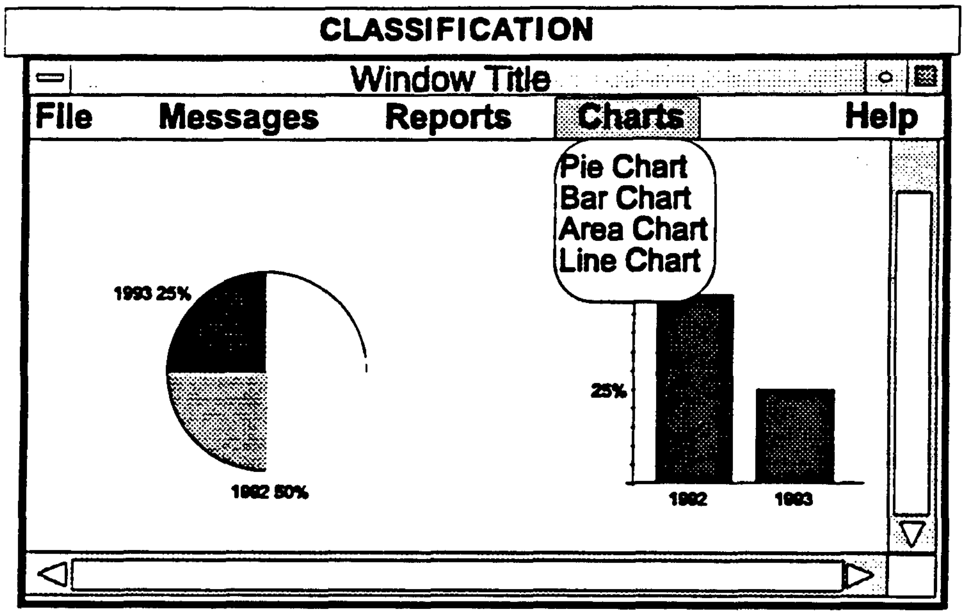
Ensure that applications adapt to individual differences and accommodate the variety of user abilities, whether novice or expert. For example, make accelerator keys for menu selection or command stacking available to the expert.

8.3.1.12 Response Demand

Demand response from the user while instructions on how to respond are still visible on the display.



To Display Required Files



To Display Specified Information

Figure 8-1. Example of How a Screen Display Can Be Customized

8.3.1.13 Consistency

- Ensure that interactive control actions are consistent in form and consequence. Employ similar means to achieve similar ends, from one transaction to the next and from one task to another throughout the command and control system.

- Ensure that results of any control entry are compatible with user expectations, so a change in the state or value of a controlled element is displayed in an expected or natural form. For example, NEXT PAGE should call up the next page of the active file, not of an unrelated file.

- When selecting names for interactive control functions, choose names that are semantically congruent with natural usage, especially for paired opposites. For example, to move a cursor up, use UP. For the opposite command, use DOWN, not LOWER.

8.3.1.14 Control

- Allow the user to complete a control entry or action through an explicit action, such as ENTER, before the system interrupts to indicate a computer-recognized word.

- Ensure that control actions are simple, particularly for real-time tasks such as fire control that require rapid user response. Control logic should permit completion of a task with the minimum number of actions, consistent with user abilities.

- Allow the user to make control entries as needed, in essence, stacking commands.

- Ensure that the sequence of control entries is not slowed by delays in computer response. In general, system response time should be in the range of 5-50 milliseconds and no longer than 0.2 seconds. System response time, in this context, refers to the time between keystroke and screen response. It does not refer to response time for a query of a database.

8.3.1.15 Feedback

- Ensure that the system provides periodic feedback to indicate that normal operation is occurring if the user waits more than 15 seconds for the computer to respond.

- Ensure that the computer acknowledges every control entry immediately; for every action by the user, some reaction from the system should be apparent.

- When computer processing is lengthy in response to a control entry, provide an overt and positive indication of when processing has been completed.

- Provide displayed feedback for all user actions; display keyed entries stroke by stroke.

8.3.1.16 Lockout

- If application processing time requires a delay of concurrent user inputs and no keyboard buffer is available, lock out the keyboard until the computer is ready to accept the next input.

- When keyboard lockout has been terminated, provide a clear indication to the user.

- In situations where control lockout occurs, provide the user with a means of aborting the transaction that caused the lockout. A method such as a special function key can accomplish this transaction. The system should not reset and lose previous processing when aborted. The system should provide the option of resetting the system.

8.3.1.17 Response Time

- Ensure that the speed of computer response to user entries is appropriate to the type of dialog. Also ensure that responses are immediate to menu selections, function keys, and most entries during graphic interaction.

- Ensure that the speed of computer response to user control entries is appropriate to the transaction involved. Generally, those transactions perceived by a user to be simple should have faster responses.

8.3.1.18 Pointer Design

- Indicate the current pointer position by displaying some distinctive pointer symbol at that point. In all cases, try to obtain the highest contrast possible between the pointer and the background. Pointer size should be such that the pointer is not lost in the clutter of the background. A contrast ratio of 3:1 is the minimum recommended for an office environment.

- Provide the user with an easy, accurate means of pointing a displayed pointer at different display elements and/or display locations. The pointer positioning should work consistently throughout the application.

- For most graphics data entry, pointing should be a dual action, first positioning a pointer at a desired position, then confirming that position to the computer.

8.3.2 Context Definition

8.3.2.1 Application-Provided Context Definition to the User

Design the interactive control of the application such that the user maintains an understanding of the context for the task being performed. Ensure that the system prompts expected user actions. For example, display the results of previous steps in the task affecting the present step and current options.

8.3.2.2 Context Established by Prior Entries

Design the interactive control software to interpret current control actions in the context of previous entries; do not require the user to re-enter data. Prompt the next logical action by the user.

8.3.2.3 Record of Prior Entries

Allow the user to request a summary of the results of prior entries (i.e., a history file) to help determine present status.

8.3.2.4 Display Operational Mode

When context for a user task is defined by the operational mode, display the current mode and any other pertinent information to the user.

8.3.2.5 Consistent Display of Context Information

Ensure information displayed to provide context for interactive control is distinctive in location and format and consistently displayed from one transaction to the next throughout all related applications.

8.3.2.6 Highlighting Selected Data

When a user is performing an operation on some selected display item, highlight that item.

8.3.2.7 Display Control Parameters

Allow the user to review any active control parameter(s).

8.3.3 Transaction Selection

8.3.3.1 Consistent CONTINUE Option

At any step in a defined transaction sequence, if there is only a single appropriate next step, provide a consistent control option, such as ENTER, to continue to the next transaction.

8.3.3.2 Indicating Control Defaults

When control is accomplished by keyed command or option code entries, and a default is defined as a null control entry, indicate that default to the user (see Figure 8-2).

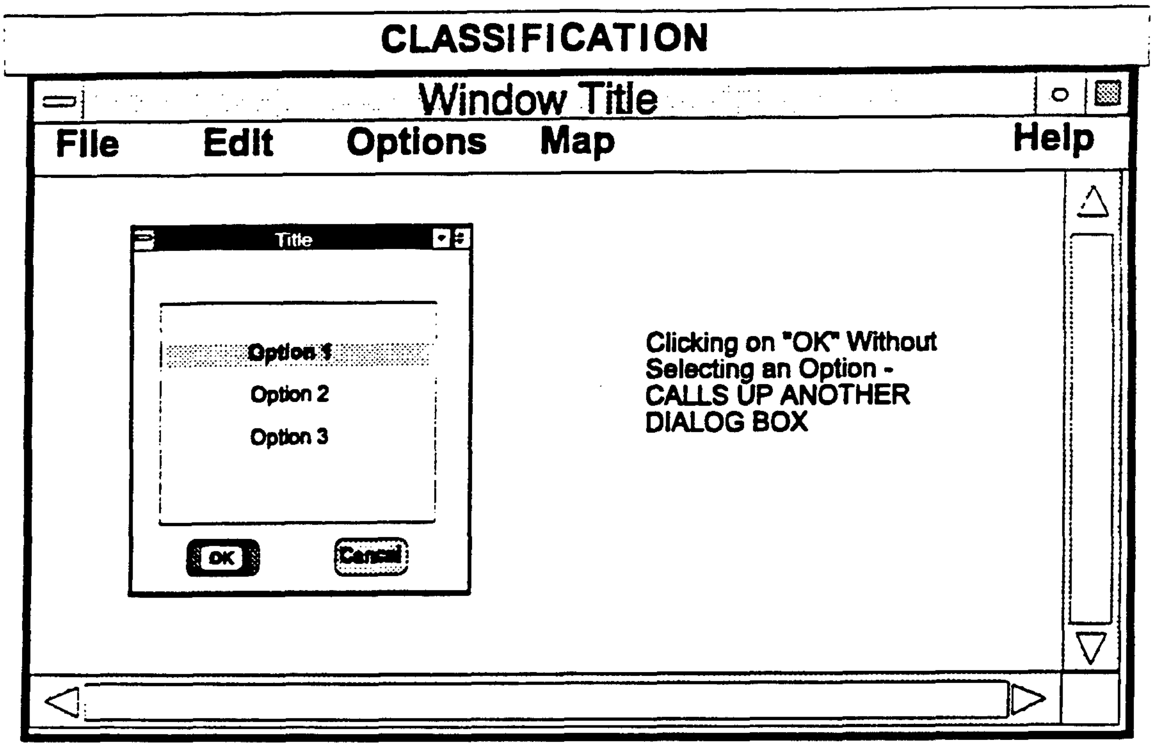


Figure 8-2. Example of How a System Can Display Default Status

8.3.3.3 User-Specified Transaction Timing

When appropriate to task requirements, allow the user to specify transaction timing. For example, the user should be able to specify when a requested transaction should start, when the transaction should be completed, and/or the periodic scheduling of repeated transactions.

8.3.3.4 Display Option Codes

When the user must select options by code entry, display the code associated with each option in a consistent, distinctive manner.

8.3.3.5 Available Options

When it is desirable not to change the menu list, ensure that the user can clearly distinguish between available and unavailable options. Displaying unavailable options in a visually distinctive manner may aid navigation.

8.3.3.6 Stacked Control Entries

Stacked commands are a function of command line interfaces. Allow the user to key a sequence of commands or option codes as a single stacked control entry. Stacked control entries, also called stacked commands, allow the user to input a series of command entries at one time. This may be done by continuous entry while the computer processes previous commands, or by typing in a series of commands and then entering them simultaneously.

- For control entry stacking, accept command names or their abbreviations or option codes, just as if those control entries had been made separately.
- Provide flexible transaction selection by allowing user to assign one name to a defined series of control entries. Use that named macro for subsequent command entry. Include a predefined informational query process (see Section 12.0) in all applications that provide a user database interface.
- For control entry stacking, require that entries be made in the order normally made when performing a succession of separate control entry actions.

8.3.3.7 Pointer Placement

- When the user must select options by keyed entry of a corresponding code, place the pointer in the control entry area at display generation.
- When the user will need to select among displayed options by pointing, place the pointer on the first (most likely) option at display generation.

8.3.3.8 Prompting Control Entries

Provide the user with whatever information may be needed to guide control entries. Incorporate prompts in a display at any point in a transaction sequence, and/or provide prompts in response to requests for HELP.

8.3.3.9 General List of Control Options

Provide a general list of basic control options that will always be available to serve as a home base or consistent starting point for control entries. For an example, see Figure 8-3 (see also Paragraph 8.2.3.3).

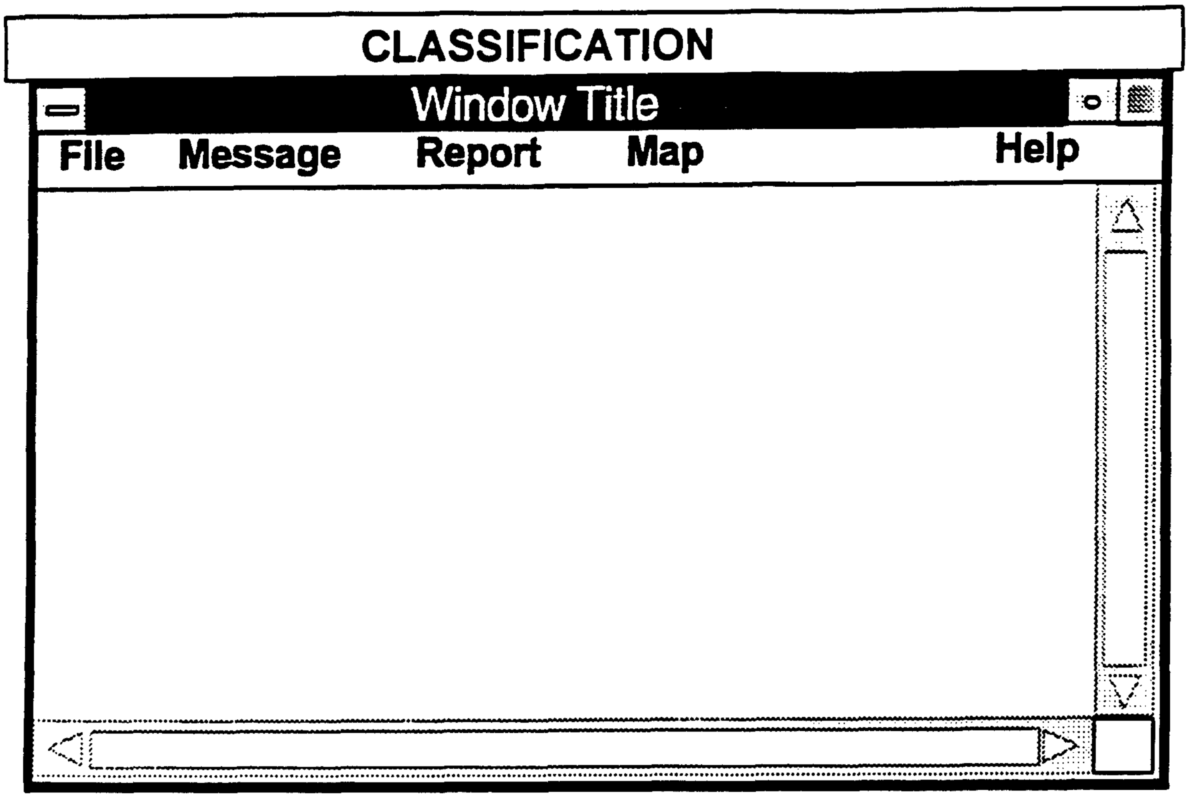


Figure 8-3. Example of a General List of Control Options

8.3.4 Interrupts

The terms that follow in caps/bold represent functions that occur in many different styles. Use of these terms should be consistent with the style upon which the application is based. Those terms listed in this section are not intended as an exclusive list of terms.

8.3.4.1 REVIEW Option

If appropriate, provide a nondestructive REVIEW option that will return to the first display in a defined transaction sequence, permitting the user to review a sequence of entries and make necessary changes.

8.3.4.2 PAUSE and CONTINUE Options

If appropriate, provide PAUSE and CONTINUE options that will interrupt and later resume a transaction sequence without any change to data entries or control logic for the interrupted transaction.

8.3.4.3 Indicating PAUSE Status

If a PAUSE option is provided, display some indication of the PAUSE status whenever that option is selected by a user, and prompt the CONTINUE action that will permit resumption of the interrupted transaction.

8.3.4.4 END Option

If appropriate, provide an END option that will conclude a repetitive transaction sequence.

8.3.4.5 Aborting or Escaping from a Function

Ensure that the system makes it easy for the user to abort, escape, or exit from a current operation or function (see also Paragraph 8.3.4.9).

8.3.4.6 Indicating System Status

Inform the user that system action is continuing. Ensure that the "working" indicator has dynamic aspects to keep the user informed of continuing system function.

8.3.4.7 SUSPEND Option

- If appropriate, provide a SUSPEND option that will preserve current transaction status when a user leaves the system and permit resumption of work when the user later logs back onto the system.

- If a SUSPEND option is provided, display some indication of the SUSPEND status whenever a user selects that option. Prompt the user with those procedures that permit resumption of the suspended transaction at the subsequent log-on. For example, specifically prompt the user with "Type EXIT to return to application."

8.3.4.8 System Interruptions

Ensure that the system interrupts the user only when necessary to prompt response, to provide essential feedback, and to signal errors.

8.3.4.9 CANCEL Option

If appropriate, provide a CANCEL option that will erase changes just made by the user and restore the current display to its previous version.

8.3.4.10 Distinctive Interrupt Options

If different types of user interrupts are provided, design each interrupt function as a separate control option with a distinct name.

8.3.4.11 GOBACK Option

If appropriate, provide a nondestructive GOBACK option that will display the previous transaction.

8.3.4.12 RESTART Option

If appropriate, provide a RESTART option that will cancel entries made in a defined transaction sequence and return to the beginning of the sequence. When data entries or changes will be nullified by restart, require the user to CONFIRM.

8.3.5 Error Management

8.3.5.1 User Confirmation of Destructive Entries

When a control entry (including log-off) will cause extensive change in stored data, procedures, and/or system operation -- particularly if it cannot be easily reversed -- notify the user and require confirmation of the action before implementing.

8.3.5.2 User Warned of Potential Data Loss

Word the prompt for a CONFIRM action to warn the user explicitly of any possible data loss.

8.3.5.3 Errors in Stacked Commands

If an error is detected in a stacked series of command entries, ensure that the system either consistently executes to the point of error or consistently requires the user to correct errors before executing any command.

8.3.5.4 Partial Execution of Stacked Commands

If only a portion of a stacked command can be executed, notify the user and provide appropriate guidance to permit correction, completion, or cancellation of the stacked command.

8.3.5.5 Flexible GOBACK for Error Correction

Allow the user to GOBACK easily to previous steps in a transaction sequence in order to correct an error or make any other desired change.

8.3.5.6 Explicit Entry of Corrections

When the user has completed correcting an error, whether a command entry or data entry, require an explicit action to re-enter the corrected material. Use the same ENTER action for re-entry that was used for the original entry.

8.3.5.7 Prompting Command Correction

If an element of a command entry is not recognized or is logically inappropriate, ensure that the system prompts the user to correct that element, rather than require re-entry of the entire command.

8.3.5.8 Immediate Data Correction

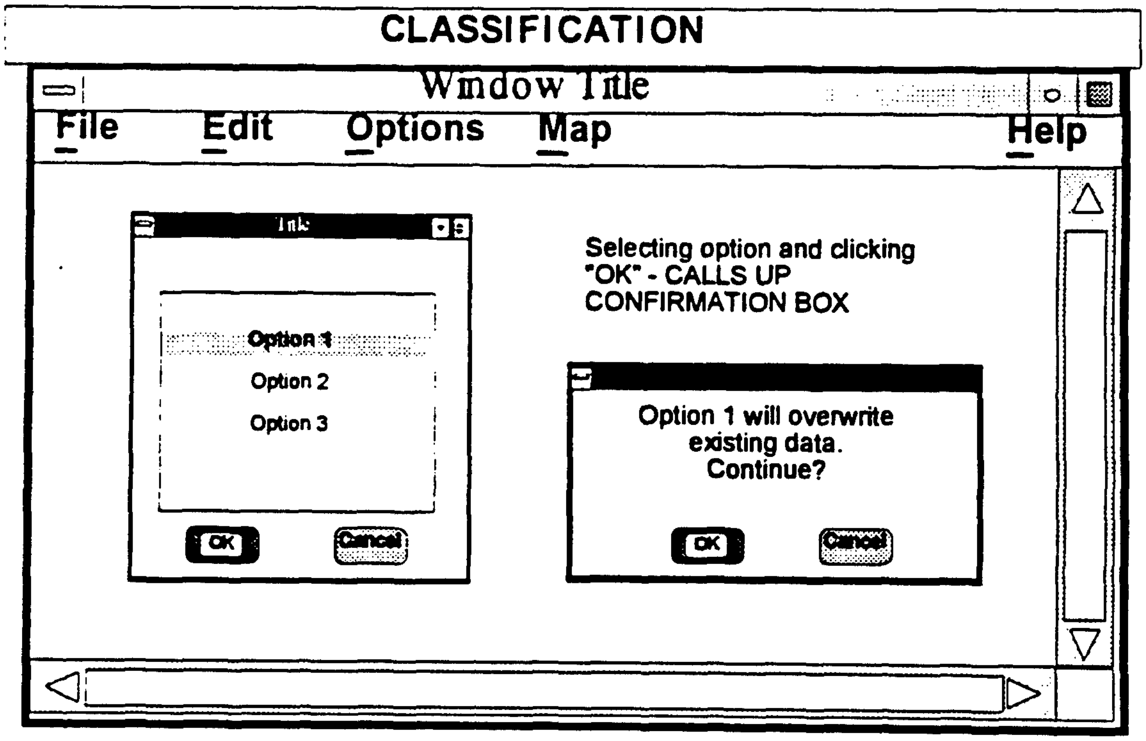
If a data entry transaction has been completed and errors are detected, allow the user to make corrections directly and immediately.

8.3.5.9 Distinctive CONFIRM Action

Provide an explicitly labeled CONFIRM control, such as a function key or widget (e.g., control button, dialog box) different from the ENTER control, for user to confirm questionable or destructive control and data entries (see Figure 8-4).

8.3.5.10 UNDO to Reverse Control Actions

Ensure any user action can be immediately reversed by an UNDO command.



Selecting option and clicking "OK" - CALLS UP CONFIRMATION BOX

Option 1 will overwrite existing data. Continue?

Figure 8-4. Example of a Distinctive Confirm Action, Using a Dialog Box

8.3.5.11 Appropriate Response to All Entries

Design software to provide an appropriate response for all possible control entries, correct and incorrect. For example, selecting an incorrect function key should cause a message indicating the appropriate selections.

8.3.5.12 Appropriate Terms for All Entries

Ensure software is consistent in the use of terms, and use only the most explicit term. Use "cancel" for cancel functions, rather than a simple acknowledgment such as "OK." Avoid complex terms (i.e., "Save & Apply" or "Exit to Prior Screen"), if possible. Ensure complex terms have one consistent meaning within an application.

8.3.5.13 Display Duration

Ensure notices, alerts, and informational displays remain visible to the user until responded to by specific user action. Field use of computers creates a situation where the user may not be continuously monitoring the screen presentation. Therefore, do not use automatic time-outs where mission-critical information is displayed.

8.3.5.14 Selection Errors

The pointing device interface uses both single and double clicks for control actions. Ensure that the software protects the system from inadvertent double clicks by the user and that the protection supplied is consistent with user and system requirements.

8.3.5.15 Inappropriate Item Selection

Cue the user to, but do not allow selection of, unavailable items. Do not allow output fields data entry without the user acknowledging selection of the option. Ensure that the software prevents data entry in any inappropriate field.

8.3.6 Alarms

8.3.6.1 Special Acknowledgment of Critical Alarms

When the user must acknowledge special or critical alarms in a unique way, such as a special combination of key strokes, ensure this acknowledgment does not inhibit or slow the response to the condition initiating the alarm.

8.3.6.2 Alarm Reset

Provide the user with a simple means of turning off an auditory alarm without erasing any displayed message that accompanies the auditory signal. For noncritical alarms, provide a simple method for acknowledging and turning off the signal.

8.3.6.3 Distinctive and Consistent Alarms

Ensure alarm signals and messages are distinctive for each class of event, such as INCOMING MESSAGE ALERT, TERMINAL STATUS, TRACK ALERT, etc.

8.3.6.4 Alarm Definition by User

When monitoring tactical situations or tactical data status, allow the user to define the conditions (e.g., priorities, percentages, target flight path, etc.) that result in a software-generated alarm, alert, or status message.

8.4 FUNCTION KEYS

The two types of function key are fixed and variable. The fixed key has only one predefined function associated with it. The variable key function will vary depending on the system mode or level within the interactive dialog. The function for the variable key is communicated to the user by changing the label located adjacent or internal to the key or through soft keys. Soft keys are objects on the display screen that represent the function keys on the keyboard. As the function of a key changes, the soft key labeling also changes. Fixed and variable function keys can be used together and with other dialog methods.

As with any interactive control method, the designer should note the following overarching design guidelines:

- Consistent design in terms of placement, labeling, and procedural logic
- Easy association with the function being called up through labeling located adjacent to the function keys
- Feedback
- Spatial consistency between the labeling and the function key.

8.4.1 General

Function keys are located on the keyboard and activate a computer software function when pressed.

8.4.1.1 Usage

Consider function key dialog for:

- Frequently required control entries

- Tasks requiring only a limited number of control entries or in conjunction with other dialog types as a ready means of accomplishing critical entries that must be made quickly, without syntax error
- Interim control entries (i.e., for control actions taken before the completion of a transaction).

8.4.1.2 Feedback for Function Key Activation

When function key activation does not result in any immediately observable response from the computer, provide users with some other form of computer acknowledgment and feedback. No system function should be activated without an indication to the user.

8.4.1.3 Disabling Unneeded Function Keys

When function keys are not needed for any current transaction, temporarily disable those keys under computer control; do not require the user to apply mechanical overlays for this purpose. (see Section 8.4.1.7).

8.4.1.4 Function Key Meaning

In general, each function key should control only one function. If a key must control more than one function, display the actual or current meaning of the function to the user by displaying soft keys on the screen.

8.4.1.5 Soft Key Design

Locate soft function keys displayed on the screen close to the actual keyboard function keys and in the same spatial orientation. For example, on command and control system keyboards with function keys across the top, place soft keys at the bottom of the screen, directly above the keyboard as illustrated in Figure 8-5.

8.4.1.6 Redundant Activation of Soft Key Function

Enable the user to activate the function represented on a soft key through either the function key or a pointing device, such as a mouse.

8.4.1.7 Indicating Active Function Keys

If some function keys are active and some are not, indicate the current subset of active keys in some noticeable way, such as brighter illumination or blanking of corresponding soft key labels on the display (see Figure 8-6).

8.4.1.8 Key Functionality Load

Avoid overloading the functionality of keys. It is recommended that no more than two functions per key be used; however, provide the user with all necessary function controls required to perform the task.

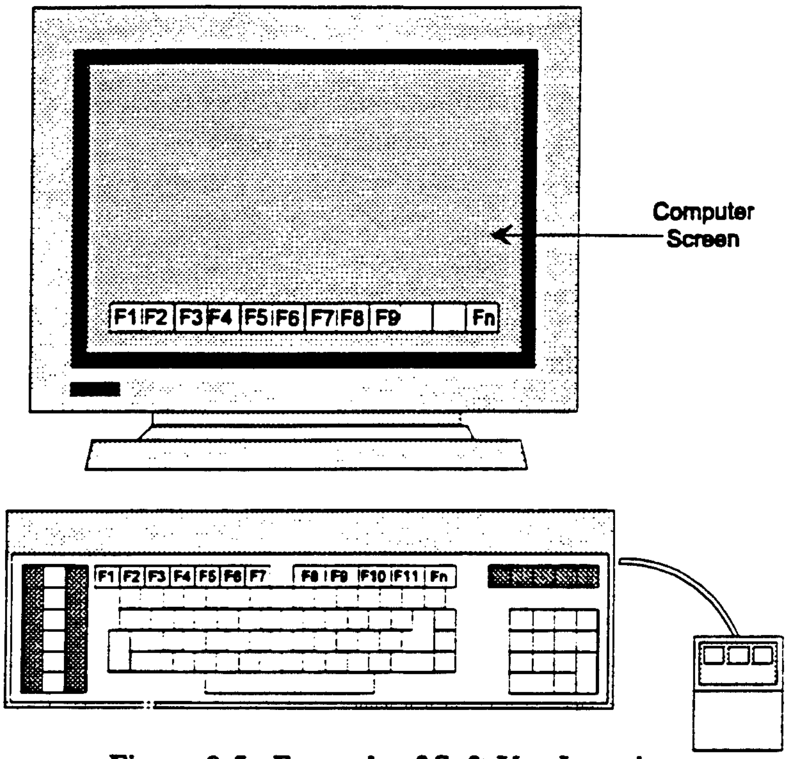

Computer Screen
F1 F2 F3 F4 F5 F6 F7 F8 F9 Fn
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 Fn

Figure 8-5. Example of Soft Key Location

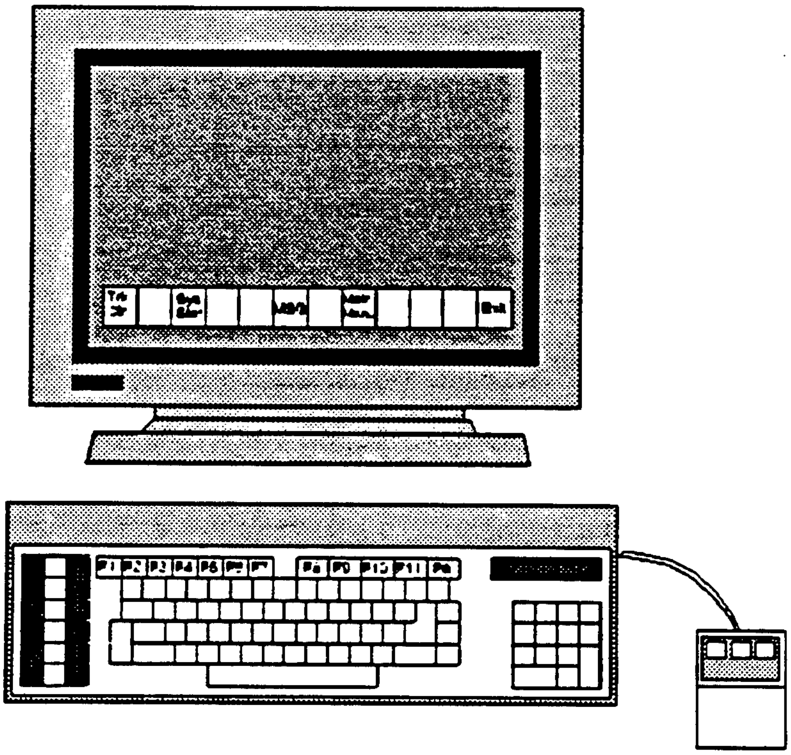


Figure 8-6. Suggested Method for Indicating Active and Inactive System Function Keys

8.4.1.9 Easy Return to Base-Level Functions

If user performs an action that changes the functions assigned to a key set, provide an easy means to return to initial, base-level functions or menu (see Figure 8-7).

8.4.2 Consistency

8.4.2.1 Consistent Functions in Different Operational Modes

When a function key performs different functions in different operational modes, assign equivalent or similar functions to the same key.

8.4.2.2 Consistent Assignment of Function Keys

If a function is assigned to a particular key in one computer transaction, assign that function to the same key in other transactions.

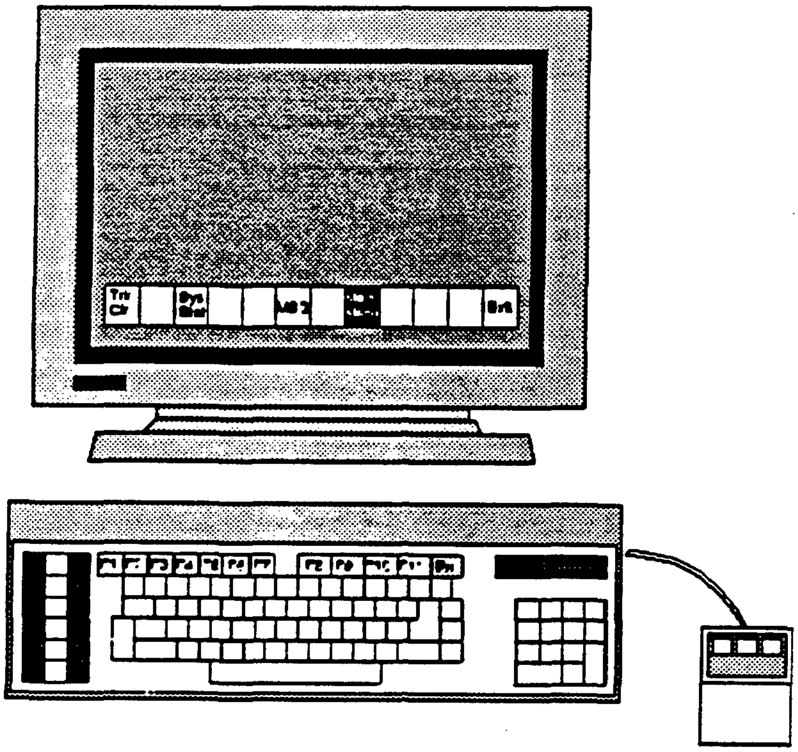


Figure 8-7. Recommended Method for a Return to Base-Level Functions

8.4.3 Double Keying

8.4.3.1 Logical Pairing of Double-Keyed Functions

If double (control/shift) keying is used, the functions paired on one key should be logically related to each other.

8.4.3.2 Consistent Logic for Double Keying

If double (control/shift) keying is used, the logical relation between shifted and unshifted functions should be consistent from one key to another.

8.4.4 Labeling

8.4.4.1 Distinctive Labeling of Function Keys

Label each function key informatively to designate the function it performs; make labels sufficiently different from one another to prevent user confusion.

8.4.4.2 Labeling Multifunction Keys

If a key is used for more than one function, always indicate to the user which function is currently available.

8.4.4.3 Labeling of Menu Options for Function Keys

When designing a command and control menu where options are selected through variable function keys, avoid using a function key number (e.g., F1, F2) as option designator. Instead, place the function key label just above the key on the display. See example in Figure 8-8.

8.4.5 Layout

8.4.5.1 Layout Compatible with Use

Make the layout of function keys compatible with their importance. Give keys for emergency functions a prominent position and distinctive coding (e.g., size and/or color).

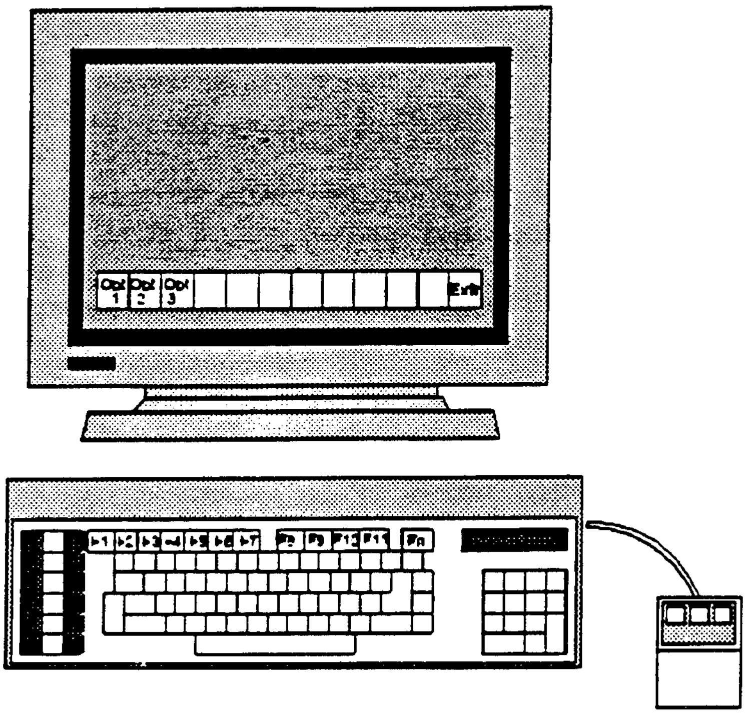
8.4.5.2 Safeguards

Provide physical protection, software disabling, or interlocks for keys with potentially disruptive consequences.

8.4.5.3 Distinctive Location

Group function keys in distinctive locations on the keyboard to facilitate learning and use. Place frequently used function keys in the most convenient locations. For command and control systems, this should be at the top of the keyboard, just below the corresponding labels.

Use This



Do Not Use This

F1 - Option 1
F2 - Option 2
F3 - Option 3

Figure 8-8. Recommended Location for Function Key Labels

8.4.6 Single Keying

8.4.6.1 Single Activation of Function Keys

Ensure that any key will perform its labeled function with a single activation and will not change function with repeated activation without indicating the new function or change in mode.

8.4.6.2 Single Key for Continuous Functions

When a function is continuously available, assign that function to a single key.

8.4.6.3 Single Keying for Frequent Functions

Keys controlling frequently used functions should allow single key action and should not require double (control/shift) keying.

REFERENCES

Paragraph	References
8.0	Smith and Mosier (1986); Helander (1986); DoD (1989b)
8.1	DoD (1992a)
8.2.1	Kearsley (1988) p. 9
8.2.4.4	Horton (1990) p. 264
8.2.4.7	Brown (1988) p. 167, para 9.27
8.2.4.8	Hurd (1983) Cited in Horton 1990, p. 264
8.2.5	Walker (1987) Cited in Horton 1990
8.2.5.2	Relles and Price (1981) Cited in Horton 1990
8.2.5.3	Relles and Price (1981) Cited in Horton 1990
8.2.5.4	Relles and Price (1981) Cited in Horton 1990
8.2.5.5	Relles and Price (1981) Cited in Horton 1990
8.2.5.6	Relles and Price (1981) Cited in Horton 1990
8.2.6	OSF (1990) p. 8-2, para 8.1.1
8.2.6.1	OSF (1990) p. 8-2, para 8.1.1
8.2.6.6	OSF (1990) p. 8-3, para 8.1.3
8.2.7	Kearsley (1988) p. 79
8.2.7.1	Kearsley (1988) p. 79
8.2.7.2	Kearsley (1988) p. 79
8.2.7.3	Kearsley (1988) p. 79
8.2.9.1	DoD (1991) p. 7-3, para 7.3
8.2.9.2	Otte (1982) p. 273, para 3.7
8.2.9.3	Smith and Mosier (1986) p. 298, para 4.0.5
8.2.9.5	Smith and Mosier (1986) p. 302, para 4.0.17
8.2.9.6	Kearsley (1988) p. 68
8.2.9.6g	Fenchel (1981) Cited in Horton (1990)
8.2.10.1	Kearsley (1988) p. 76

REFERENCES (cont'd)

Paragraph	References
8.2.10.2	Unix System Laboratories (1991) p. 4-27, para 3
8.2.12.1	Kearsley (1988) p. 25
8.2.13.1	Unix System Laboratories (1991) p. 4-27, para 3
8.2.13.7	Nickerson (1986); Smith and Mosier (1986) p. 302, para 4.0.17
8.3.1.1	Smith and Mosier (1986) para 3.0-9
8.3.1.2	Lickteig (1989) p. 9
8.3.1.3	Williams et al. (1987a) Appendix A p. A-2; Smith and Mosier (1986) para 8.3.7.2-2
8.3.1.4	Smith and Mosier (1986) para 3.0-12
8.3.1.5	Chao (1987) p. 360 and 361
8.3.1.6	Smith and Mosier (1986) para 3.0-17
8.3.1.7	Smith and Mosier (1986) para 3.0-7
8.3.1.8	Lickteig (1989) p. 35
8.3.1.9	Lickteig (1989) p. 34; Williams et al. (1987a) Appendix A p. A-3
8.3.1.10	Smith and Mosier (1986) para 3.0-8
8.3.1.11	Hamel and Clark (1986) p. 29; Smith and Mosier (1986) para 3.0-3
8.3.1.12	Williams et al. (1987b) Appendix A p.A-3
8.3.1.13a	Smith and Mosier (1986) para 3.0-6, 19
8.3.1.13b	Smith and Mosier (1986) para 3.0-16
8.3.1.13c	Smith and Mosier (1986) para 3.0-11
8.3.1.14a	Smith and Mosier (1986) para 3.0-5
8.3.1.14b	Smith and Mosier (1986) para 3.0-2
8.3.1.14c	Smith and Mosier (1986) para 3.0-19
8.3.1.14d	Smith and Mosier (1986) para 3.0-19; Salvendy (1987)
8.3.1.15a	Williams et al. (1987b) Appendix A p. A-3
8.3.1.15b	Hamel and Clark (1986) 3.0-20; Smith and Mosier (1986) para 3.0

REFERENCES (cont'd)

Paragraph	References
8.3.1.15c	Smith and Mosier (1986) para 3.0-15; Baeker (1980); Hamel and Clark (1986) p. 30; Mallary (1985) p. 26; McCann (1983) p. 4; Slominski and Young (1988) p. 5
8.3.1.15d	Smith (1986) para 1.0-3
8.3.1.16a	DoD (1989a) para 5.15.4.1.1.1; Smith and Mosier (1986) para (1986) p. 30; DoD (1989b)
8.3.1.16b	DoD (1989a) para 5.15.4.1.1.2
8.3.1.16c	Smith and Mosier (1986) para 3.0-21; DoD (1989a) para 5.15.4.1.1.3
8.3.1.17a	Smith and Mosier (1986) para 3.1-2
8.3.1.17b	Smith and Mosier (1986) para 3.0-18;
8.3.1.18a	Bowser (1991) p. 6; Lewis and Fallesen (1989) p. 94; HFS (1988)
8.3.1.18b	Bowser (1991) p. 6; Lewis and Fallesen (1989) p. 94
8.3.1.18c	Lewis and Fallesen (1989) p. 94
8.3.2.1	Bowser (1991) p. 6; Smith and Mosier (1986) para 3.4-1; Bullinger et al. (1987) pp. 312-3; Hamel and Clark (1986) p. 26-27; Mallary (1985) p. 28, 9 p. 2, 14 p. 6, 3 p. 360
8.3.2.2	Smith and Mosier (1986) para 3.4-2
8.3.2.3	Smith and Mosier (1986) para 3.4-3
8.3.2.4	Smith and Mosier (1986) para 3.4-4
8.3.2.5	Smith and Mosier (1986) para 3.4-7
8.3.2.6	Smith and Mosier (1986) para 3.4-6
8.3.2.7	Smith and Mosier (1986) para 3.4-5
8.3.3.1	Smith and Mosier (1986) para 3.2-12
8.3.3.2	Smith and Mosier (1986) para 3.2-11
8.3.3.3	Smith and Mosier (1986) para 3.2-19
8.3.3.4	Smith and Mosier (1986) para 3.2-8
8.3.3.5	Smith and Mosier (1986) para 3.2-10

REFERENCES (cont'd)

Paragraph	References
8.3.3.6	Smith and Mosier (1986) para 3.2-13
8.3.3.6a	Smith and Mosier (1986) para 3.2-15
8.3.3.6b	Bowser (1991) p. 6; Smith and Mosier (1986) para 3.2-18
8.3.3.6c	Smith and Mosier (1986) para 3.2-14
8.3.3.7a	Smith and Mosier (1986) para 3.2-7
8.3.3.7b	Smith and Mosier (1986) para 3.2-6
8.3.3.8	Smith and Mosier (1986) para 3.2-5
8.3.3.9	Smith and Mosier (1986) para 3.2-2
8.3.4.1	Smith and Mosier (1986) para 3.3-5
8.3.4.2	Smith and Mosier (1986) para 3.3-8
8.3.4.3	Bowser (1991) p. 7; Smith and Mosier (1986) para 3.3-7
8.3.4.4	Baeker (1980); Williams et al. (1987a) Appendix A p. A-2; Smith and Mosier (1986) para 3.3-1
8.3.4.6	Harrell (1987) p.5
8.3.4.7	Smith and Mosier (1986) para 3.3-3
8.3.4.8	Smith and Mosier (1986) para 3.3-2
8.3.4.9	Smith and Mosier (1986) para 3.3-4
8.3.4.10	Smith and Mosier (1986) para 3.3-6
8.3.5.1	Smith and Mosier (1986) para 3.5-7 and 3.5-11
8.3.5.2	Smith and Mosier (1986) para 3.5-8
8.3.5.3	Smith and Mosier (1986) para 3.5-4
8.3.5.4	Smith and Mosier (1986) para 3.5-5
8.3.5.5	Smith and Mosier (1986) para 3.5-13
8.3.5.6	Smith and Mosier (1986) para 3.5-6
8.3.5.7	Smith and Mosier (1986) para 3.5-3
8.3.5.8	Smith and Mosier (1986) para 3.5-12
8.3.5.9	Smith and Mosier (1986) para 3.5-9

REFERENCES (cont'd)

Paragraph	References
8.3.5.10	Smith and Mosier (1986) para 3.5-10
8.3.5.11	Smith and Mosier (1986) para 3.5-1
8.3.5.12	Bowser (1991) p. 8
8.3.5.13	Bowser (1991) p. 7
8.3.6.1	Smith and Mosier (1986) para 3.6-5
8.3.6.2	Smith and Mosier (1986) para 3.6-4 and 3.6-3
8.3.6.3	Smith and Mosier (1986) para 3.6-2
8.3.6.4	Smith and Mosier (1986) para 3.6-1
8.4.1.1a	Smith and Mosier (1986) para 3.1.4-2
8.4.1.1b	Smith and Mosier (1986) para 3.1.4-1
8.4.1.1c	Smith and Mosier (1986) para 3.1.4-3
8.4.1.2	Bowser (1991) p. 9; Smith and Mosier (1986) para 3.1.4-10
8.4.1.3	Smith and Mosier (1986) para 3.1.4-12
8.4.1.4	Ziegler and Fähnrich (1988) p. 129
8.4.1.5	Ziegler and Fähnrich (1988) p. 129
8.4.1.6	Ziegler and Fähnrich (1988) p. 129
8.4.1.7	Smith and Mosier (1986) para 3.1.4-11
8.4.1.8	Nielsen (1987) p. 248; Bullinger et al. (1987) p. 309; McCann (1983) pp.3-4
8.4.1.9	Smith and Mosier (1986) para 3.1.4-16
8.4.2.1	Smith and Mosier (1986) para 3.1.4-15
8.4.2.2	Smith and Mosier (1986) para 3.1.4-14
8.4.3.1	Smith and Mosier (1986) para 3.1.4-7
8.4.3.2	Smith and Mosier (1986) para 3.1.4-8
8.4.4.1	Smith and Mosier (1986) para 3.1.4-4
8.4.4.2	Smith and Mosier (1986) para 3.1.4-5
8.4.4.3	Sidorsky (1994) p. 1.1-12

REFERENCES (cont'd)

Paragraph	References
8.4.5.1	Smith and Mosier (1986) para 3.1.4-18
8.4.5.2	Smith and Mosier (1986) para 3.1.4-18
8.4.5.3	Smith and Mosier (1986) para 3.1.4-17
8.4.6.1	Smith and Mosier (1986) para 3.1.4-9
8.4.6.2	Smith and Mosier (1986) para 3.1.4-13
8.4.6.3	Smith and Mosier (1986) para 3.1.4

9.0 TEXT

Two topics will be addressed in Section 9.0. Subsection 9.1 addresses general topics unique to textual windows (i.e., data entry/update screens) that were not covered in Section 5.0, Windows. Subsection 9.2 addresses form filling as an interactive dialog. This approach to data entry requires little or no training and allows a relatively slow system response time. Applications primarily use form filling for completing standard message and data entry forms. As with any aspect of the HCI design, consistency of design is of paramount importance. The guidelines presented deal more with interactive control than with data entry. For more information on data entry, see Smith and Mosier (1986), MIL-STD-1472D (DoD 1989a), or DOD-HDBK-761A (DoD 1989b).

9.1 TEXTUAL WINDOWS

This section addresses general guidelines related to windows that are primarily textual (i.e., data entry/update screens).

9.1.1 Data Field Labeling

In general, the appearance of the data should be pleasing to the eye with the arrangement uncluttered and functionality efficient. The following list of guidelines should help to achieve these objectives:

- Ensure that displays are not different from paper forms without justification; field ordering should be in logical sequence from the user's point of view.

- Ensure that the layout of data fields is consistent within an application, because one of the overall DoD architecture goals is to have consistency across all DoD applications in the layout of commonly used display (e.g., the "views" presented by applications for querying related databases).

- Ensure that data field labels are easily distinguishable from actual data. This distinction may be achieved using different fonts for labels and data or special characters as separators. For example, each label should be followed by a colon (:) and separated from the actual data by at least two spaces.

- Distinctly separate columnar data (at least three spaces between columns), with column headings displayed above the data and at least one row separating the column heading and the data.

- Ensure that labels are consistent throughout an application or set of applications.

- In ordinary use, ensure that field labels are protected and transparent to keyboard control so the cursor skips over them when spacing or tabbing.

- When a dimensional unit (e.g., $) is always associated with a field, display it as part of the label so entry is not required by the user.

9.1.2 Updatable Fields

Guidelines for data field updates follow:

- Distinguish updatable fields by underscores below the data field. If highlights or colors are also used, they should be the same throughout an application or set of applications.

- Cues should distinguish required from optional fields and should be consistent throughout an application or set of applications.

- When the length of a field is variable, the user should not have to right- or left-justify or remove blanks from the entered data.

- Ensure that the user is able to enter data in familiar units. The application should perform any required conversions (e.g., between geographic, geodetic, and Military Grid Reference System coordinates).

- Authorized personnel should be able to selectively inhibit updatable fields in a multi-field display. Such a feature would allow trainees to take on increasing database maintenance responsibilities as they learn. It also supports efficient on-line accomplishments of "mass changes" when batch updates are not available.

9.1.3 Text Cursor

The purpose of the text cursor is to indicate to the user where entered data will be placed. The text cursor can be in any updatable input field. Guidelines for the text cursor follow:

- If the user clicks on a non-updatable field or anywhere on the form, the text cursor should not move.

- The text cursor should move between and within fields with the mouse or by using the Return/Enter key, Tab key, or the arrow keys.

- The cursor should not obscure the character displayed in the position it designates except for password and other non-display fields.

- When in insert mode, the text cursor should appear between the characters where the inserted text will be placed.

- When in overwrite mode, indicate (e.g., on status bar) that the current status of the application is in overwrite mode, or the text cursor should highlight the character that will be replaced.

9.2 FORM FILLING

9.2.1 General

Form filling is the method of interaction where the user enters a series of commands or data items in predefined, mandatory or optional fields.

9.2.1.1 Usage

- Use form-filling dialog as an aid for composing complex control entries.
- Use form-filling dialog as a means of displaying default values for the parameters in complex control entries.
- Use form-filling dialog for tasks where flexibility in data entry is needed (such as the inclusion of optional as well as required items), where users will have moderate training, and/or where computer response may be slow.

9.2.1.2 Interrupts for Multiple Entries

Where forms have multiple entries, provide the user GOBACK, CANCEL, and RESTART capabilities for editing the form prior to final input into the system (see Figure 9-1).

9.2.1.3 Explicit Data Entry

Data entry should be accomplished through an explicit action, such as pressing the ENTER key.

9.2.2 Defaults

9.2.2.1 Automatic Display of Default Data

If default values are used in data entry fields, display them automatically in the appropriate data entry field.

9.2.2.2 Replacement of Default Values

When the user replaces a default value in a data entry field, ensure that the default definition is not changed.

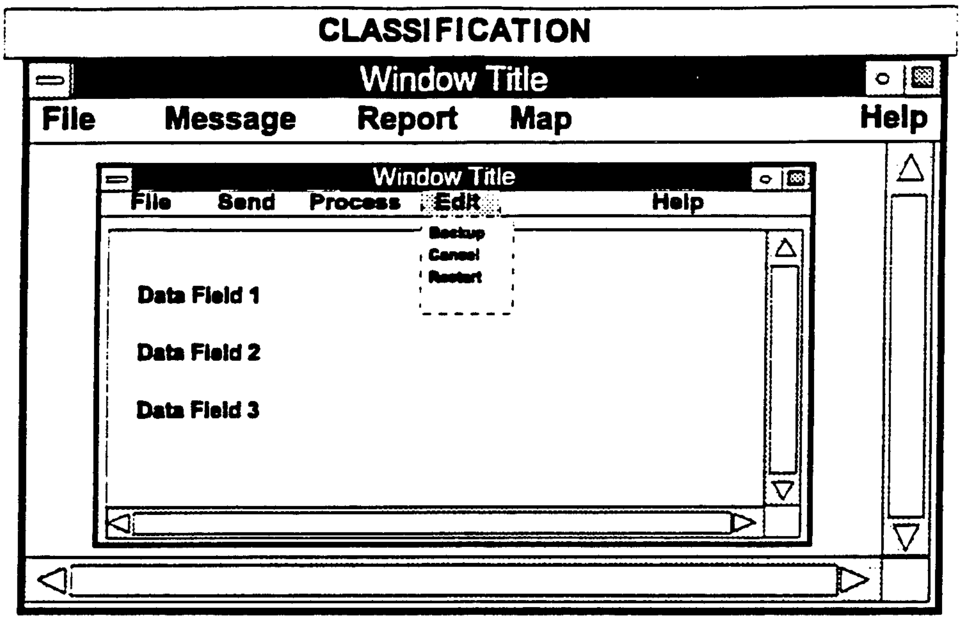


Figure 9-1. Example of Interrupt Capability for Multiple Entries

9.2.3 Consistency

9.2.3.1 Consistent Format for Control Forms

Ensure that forms for control entry are consistent in format.

9.2.3.2 Format of Form and Hard Copy

When the user enters data from hard copy into a computer, where possible, the computer form and hard-copy format should be identical (see Figure 9-2).

9.2.3.3 Entry Dialog Consistency

Dialog strategies for entering words and numbers should be consistent for a given set of logical functions throughout the system.

9.2.3.4 Standard Formats

Data and/or processes that have standard information requirements need to provide the standard format as part of the data screen. Message formats should include a template for the standard format. Using data entry screens that do not conform to user-accepted format will confuse users.

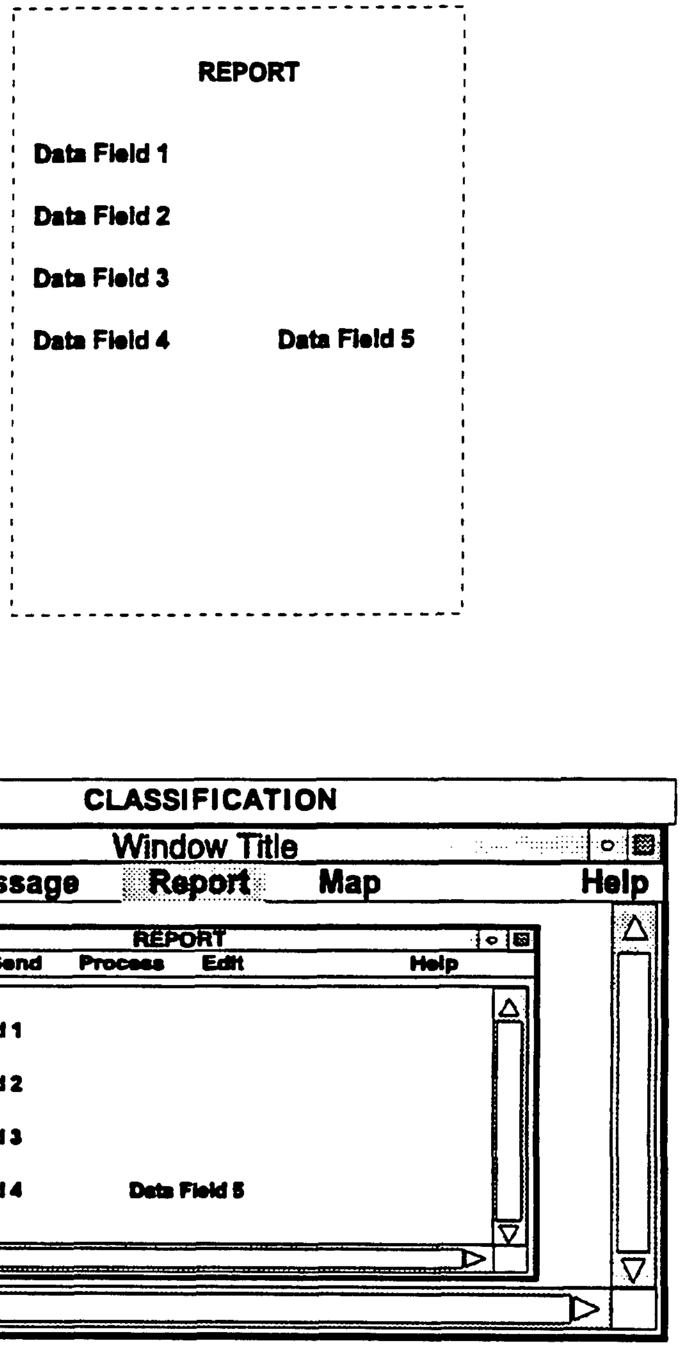


Figure 9-2. Example of How a Paper Entry Form and a Computer Data Entry Form Should Be Consistent

9.2.4 Cursor Movement

9.2.4.1 Cursor Movement Into Non-Data Area

Applications should not allow the user to move the cursor into a non-data-entry area during form filling.

9.2.4.2 Convenient Cursor Movement

Ensure that the user has a convenient method for cursor control, such as the use of the tab or pointing device.

9.2.4.3 Cursor Movement by Explicit Action

When moving from one data entry field to another, require the user to take an explicit action, such as hitting the tab control. The software should not automatically advance to the next field.

9.2.4.4 Cursor/Pointing Device Interaction

Pointing device-to-cursor movement ratio should be close to 1:1. If appropriate, the user should be able to select the movement ratio.

9.2.4.5 Initial Cursor Location

When the user first calls up a form, the cursor should be positioned in the first character space of the first data entry field (see Figure 9-3).

9.2.5 Data Field

9.2.5.1 Variable Data Field Format

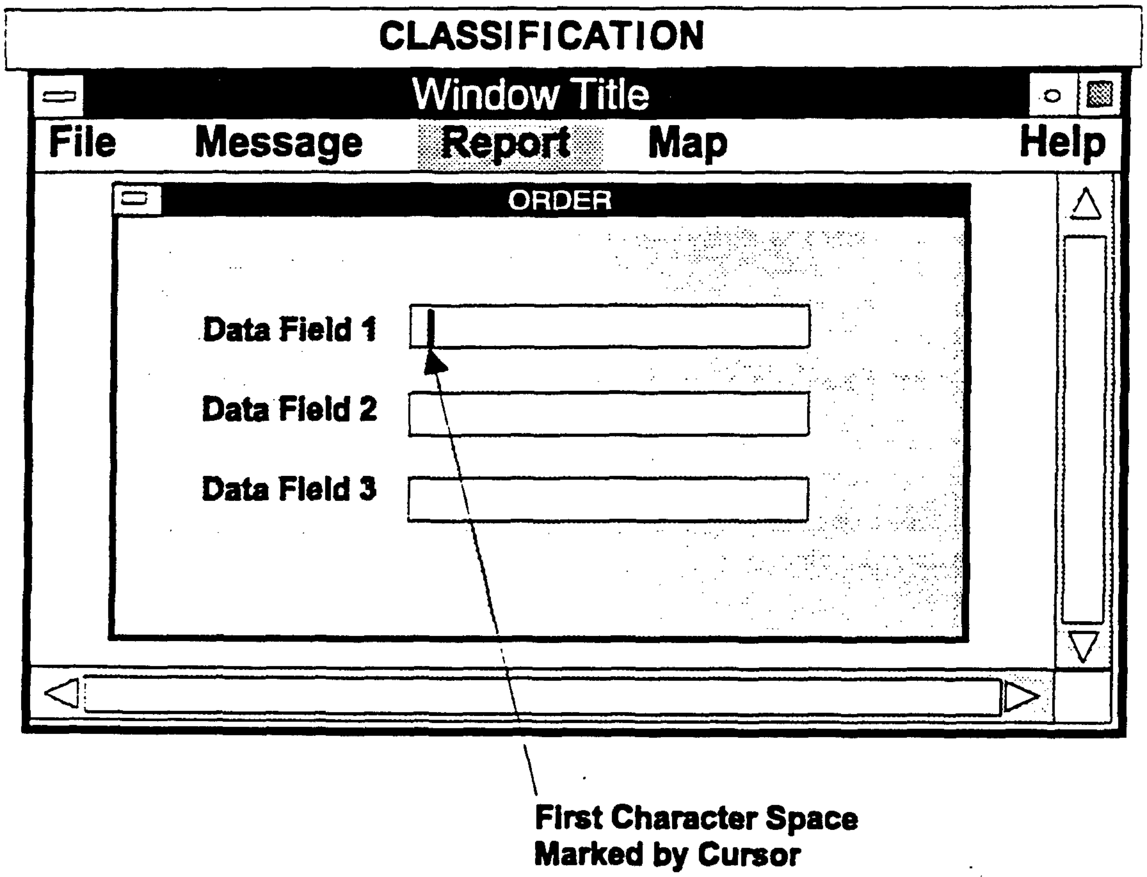
For data entry fields with variable lengths, the software should automatically justify or truncate the data for the user. No leading characters should be required. See the example in Figure 9-4.

9.2.5.2 Consistent Format of Data Fields

The format of data fields used frequently on different forms within and among applications should be consistent from one display to another and should use a format convention consistent with the user's expectations.

9.2.5.3 Subgroups Within a Data Field

For data fields longer than 5 to 7 characters, break the field into subgroups of 3 to 4 characters that are separated by a space or delimiter. This should follow a convention consistent with the user's expectations (i.e., names, addresses, some descriptive information should not be subdivided).



First Character Space
Marked by Cursor

Figure 9-3. Cursor Should Appear in First Character Space of First Data Entry Field

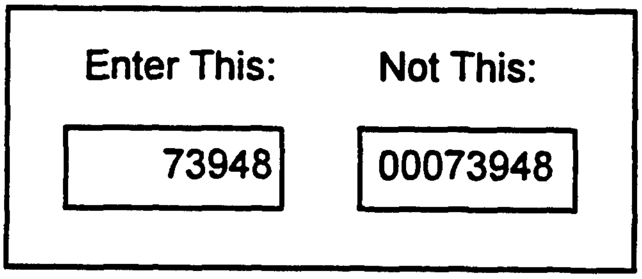


Figure 9-4. Data Entry Should Not Require Leading Zeros

9.2.5.4 Data Field Boundaries

Data fields should have distinctly marked boundaries.

9.2.5.5 Data Field Identification

Data entry fields should be clearly identified. Because the interface designs are often complex, users need a positive visual means to identify data entry fields.

9.2.5.6 Field Length

Data entry fields should be of fixed length, with cues given for their length (see Figure 9-5).

9.2.5.7 Overwriting

Data entry should not require overwriting of existing or default information. The field should either be empty, or the user should be required to perform an explicit control entry to erase the default data.

9.2.5.8 Numeric Data Fields

Numeric data in decimal format should use the decimal as part of the data display. Care should be taken to ensure the field size is adequate for the data range.

9.2.6 Error Management

9.2.6.1 Error Correction for Characters and Fields

Ensure that the user can easily correct errors on a character-by-character and field-by-field basis.

9.2.6.2 Error Messages

Ensure that the software provides understandable error messages to the user when an unacceptable value is entered in a data field.

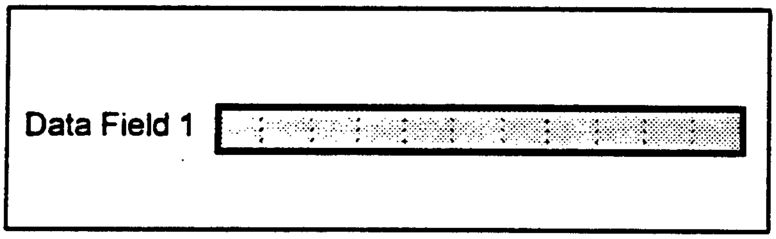


Figure 9-5. Visual Cues for Field Length

9.2.7 Form Layout

9.2.7.1 Multiscreen Form Numbering

If multiscreens are used for a transaction, provide page numbers for each screen. Also provide a means for rapidly returning to any page.

9.2.7.2 Logical Grouping of Data Fields

Group related data fields together on the same form.

9.2.7.3 Explanatory Messages for Data Fields

Provide explanatory messages for data fields that become visible when the cursor is placed in a field, when a user queries a field by clicking on the title, or by a context-sensitive help system. See the example in Figure 9-6.

9.2.7.4 Distinguishing Data Fields from Other Information

Distinguish messages and instructions on a form from data entry fields by means of consistent location or other means of highlighting (see Figure 9-6).

9.2.7.5 Spacing and Boundaries

Ensure that each data field has visible space and boundaries between it and other fields.

9.2.7.6 Grouping and Sequencing Fields

Group and order data entry fields should be grouped and ordered on the form in a way that is logical for the task to be performed. This can be by sequence, frequency, or importance.

9.2.7.7 Form Title

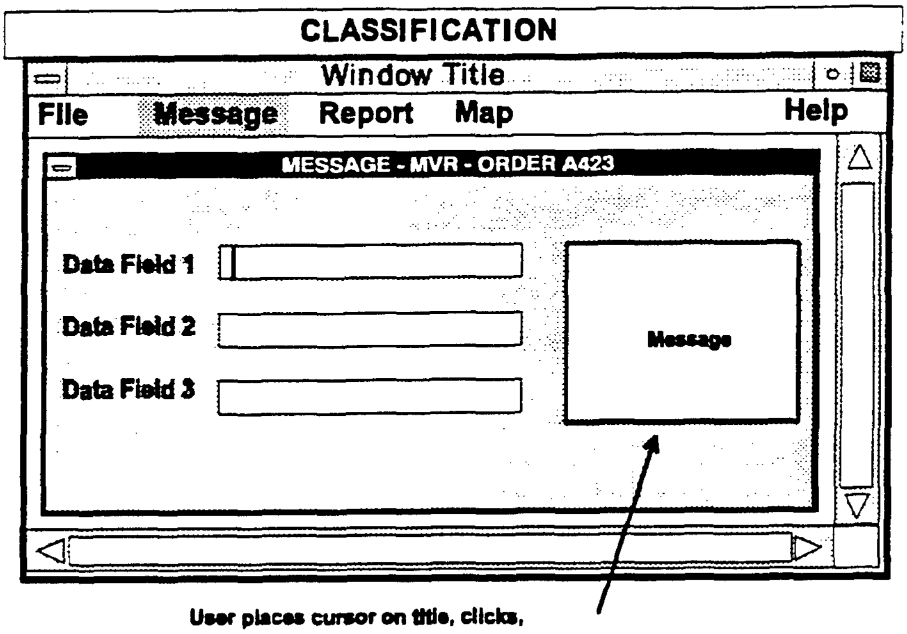
Ensure that each form-filling dialog display page has a meaningful title located at the top of the form, as illustrated in Figure 9-7.

9.2.7.8 Optional Field Labels

Optional fields should be labeled or coded in a readily apparent manner (see Figure 9-8).

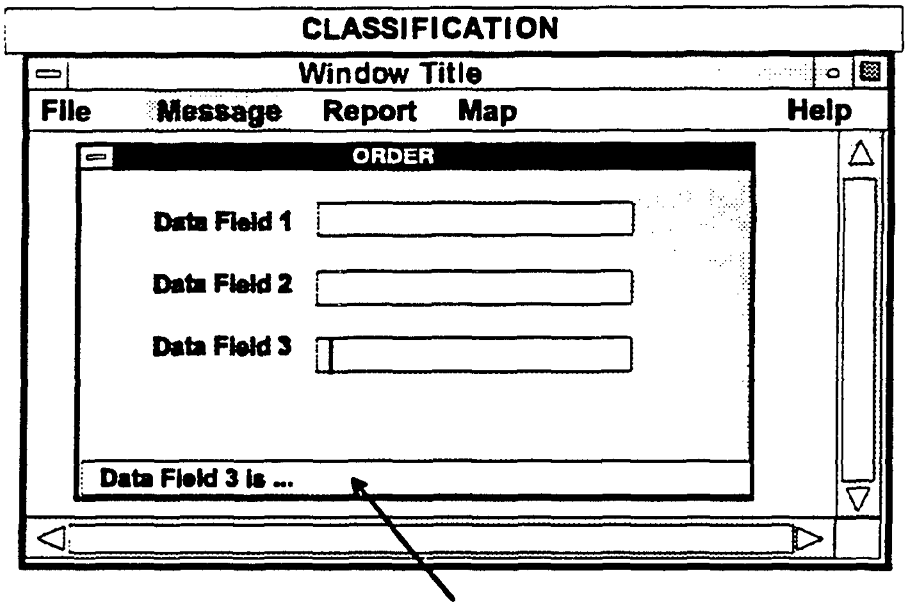
9.2.7.9 Optional Field Defaults

When a data entry field in a form is optional, any value displayed in that field should be a default value.



User places cursor on title, clicks, information on field appears as a pop-up subwindow

OR



Data Field 3 is ...

Information on data field appears automatically when cursor is placed in field.

Figure 9-6. Example of How Explanatory Messages Can Be Provided

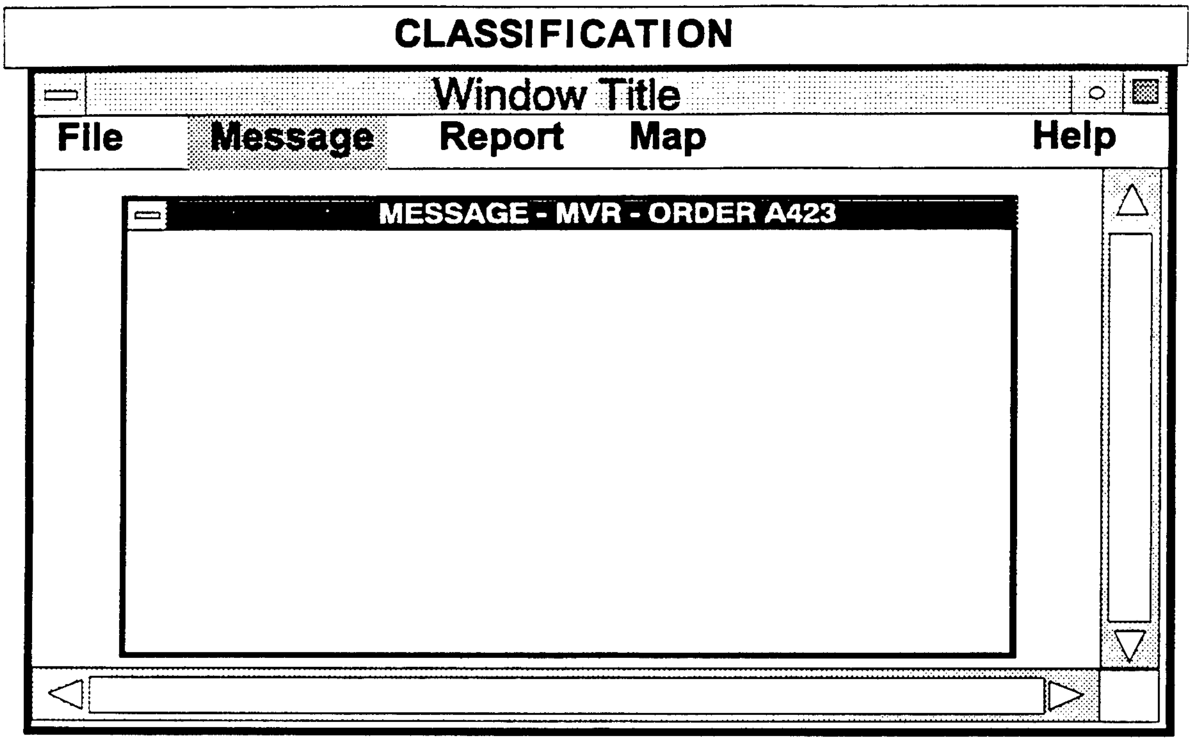


Figure 9-7. Example of a Form Title

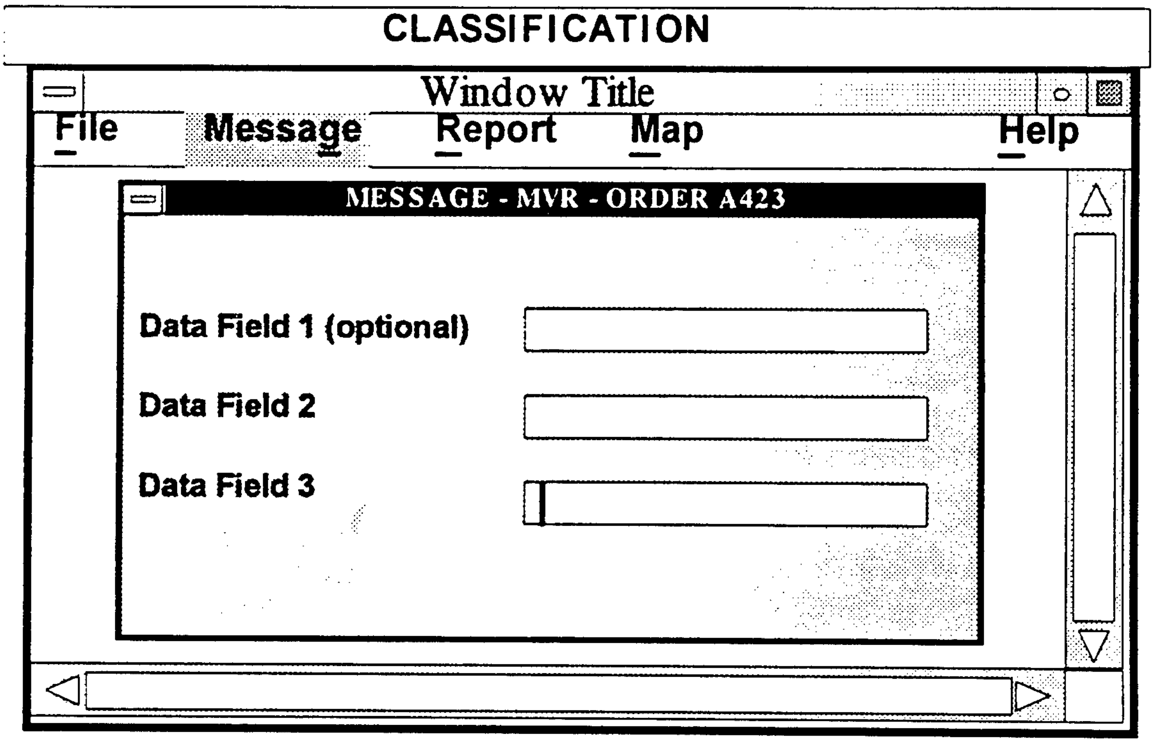


Figure 9-8. Example of an Indication of an Optional Field

9.2.7.10 Mandatory Fields

The software application should not allow the user to bypass a mandatory field without data entry (see Paragraph 9.2.5.3).

9.2.8 Labeling

9.2.8.1 Distinctive Labeling

Data fields, unless similar or identical, should have distinctive, explicitly descriptive labels.

9.2.8.2 Data Field Label Location

Application data entry field labels should be located either directly to the left or above the actual entry field and separated by at least one character.

9.2.8.3 Similar Data Field Labeling

Similar data entry fields should be labeled and located consistently for all forms.

9.2.8.4 Consistent Labels

Labels and instructions should be consistent from one application to another within related applications and to the extent possible across all systems.

9.2.8.5 Field Label Familiarity

Labels for data fields should be composed of terms familiar to the user and the task to be performed.

9.2.8.6 Understandable Labeling

Labeling for data fields and instructions should be easily understood by the typical user.

9.2.8.7 Units of Measure

Units of measure should be part of the data entry field label. If measurement units can change, this portion of the label should change automatically when new units are selected.

9.2.8.8 Blanks Versus Nulls

There should be a visible distinction between blanks and nulls in a data field.

REFERENCES

Paragraph	References
9.1	DoD (1992a)
9.2.1.1a	Smith and Mosier (1986) para 3.1.2-2
9.2.1.1b	Smith and Mosier (1986) para 3.1.2-3
9.2.1.1c	Smith and Mosier (1986) para 3.1.2-1
9.2.1.2	Chao (1986) p. 13
9.2.1.3	Chao (1986) p. 13
9.2.2	Chao (1986) p. 13
9.2.3.1	Smith and Mosier (1986) para 3.1.2-4
9.2.3.2	Sidorsky (1984) p. 1.1-11; Chao (1986) p. 13
9.2.3.3	Lewis and Fallesen (1989) p. 8; Chao (1986) p. 13
9.2.3.4	Bowser (1991) p. 8
9.2.4.1	Chao (1986) p. 13
9.2.4.2	Shneiderman (1988) p. 702
9.2.4.3	Chao (1986) p. 13
9.2.4.4	Chao (1986) p. 13
9.2.5	Chao (1986) p. 13
9.2.5.9	Bowser (1991) p. 8
9.2.6	Shneiderman (1988) p. 702
9.2.7.1	Chao (1986) p. 13
9.2.7.2	Chao (1986) p. 13
9.2.7.3	Chao (1986) p. 703
9.2.7.4	Chao (1986) p. 13
9.2.7.5	Shneiderman (1988) p. 702
9.2.7.6	Shneiderman (1988) p. 702
9.2.7.7	Shneiderman (1988) p. 702
9.2.7.8	Shneiderman (1988) p. 703; Chao (1986) p. 13

REFERENCES (cont'd)

Paragraph	References
9.2.7.9	Chao (1986) p. 13
9.2.7.10	Avery et al. (1990) p. 3-19
9.2.8.1	Chao (1986) p. 13
9.2.8.2	Chao (1986) p. 13
9.2.8.3	Chao (1986) p. 13
9.2.8.4	Shneiderman (1988) p. 702
9.2.8.5	Shneiderman (1988) p. 702
9.2.8.6	Shneiderman (1988) p. 702
9.2.8.7	Chao (1986) p. 13
9.2.8.8	Chao (1986) p. 13

10.0 GRAPHICS

Graphical presentation of data is a critical feature of many emerging DoD applications. This section provides guidelines for presenting data in graphical formats. The applications discussed here include tactical graphics (overlays, symbology, and terrain representation), pictographic representations (digitized maps, pictures, etc.), and presentation graphics (charts and graphs). Guidelines pertaining to graphical characteristics of the user interface (e.g., screen design, windows, icons, buttons, etc.) are presented in other sections of this document.

Most of the guidelines presented in this section were obtained from Lewis and Fallesen (1989) and Smith and Mosier (1986), who included information gathered from relevant Military Standards and other key documents. Additional guideline materials were obtained through literature reviews.

Subsection 10.1 focuses on map graphics. The designer of map graphic displays should note the following overarching guidelines that are relevant to electronic map displays:

- The design of maps, including the use of symbology, should be consistent with the user's expectations.

- The level of detail should be consistent with the operational need. Too much or too little detail limits the usefulness of the map.

- Map graphics should have tools built in that allow the user to move easily around the map, to include zooming, panning, insets, registration, and keys for scale.

Subsection 10.2 focuses on presentation graphics. The goal of presentation graphics is to communicate effectively to the user. The idea, information, or concept communicated should be clear and unambiguous when presented in a visual form; otherwise alternate communication modes should be used. Some emerging technological capabilities allow the direct manipulation of elements of graphic objects within an application. These capabilities should be included in applications designed to interactively create graphics. The *Style Guide* will address three aspects of presentation graphics: graphs, pictures, and diagrams.

10.1 MAPS AND SITUATION DISPLAYS

10.1.1 General

Maps refer to projected representations of geographic data, usually on flat surface displays. Maps include both natural and man-made features and text and/or graphics and colors used to describe or code those features. Situation displays provide a means of relating changing conditions or events to geographic features represented on maps. Figure 10-1 illustrates a typical map graphic display.

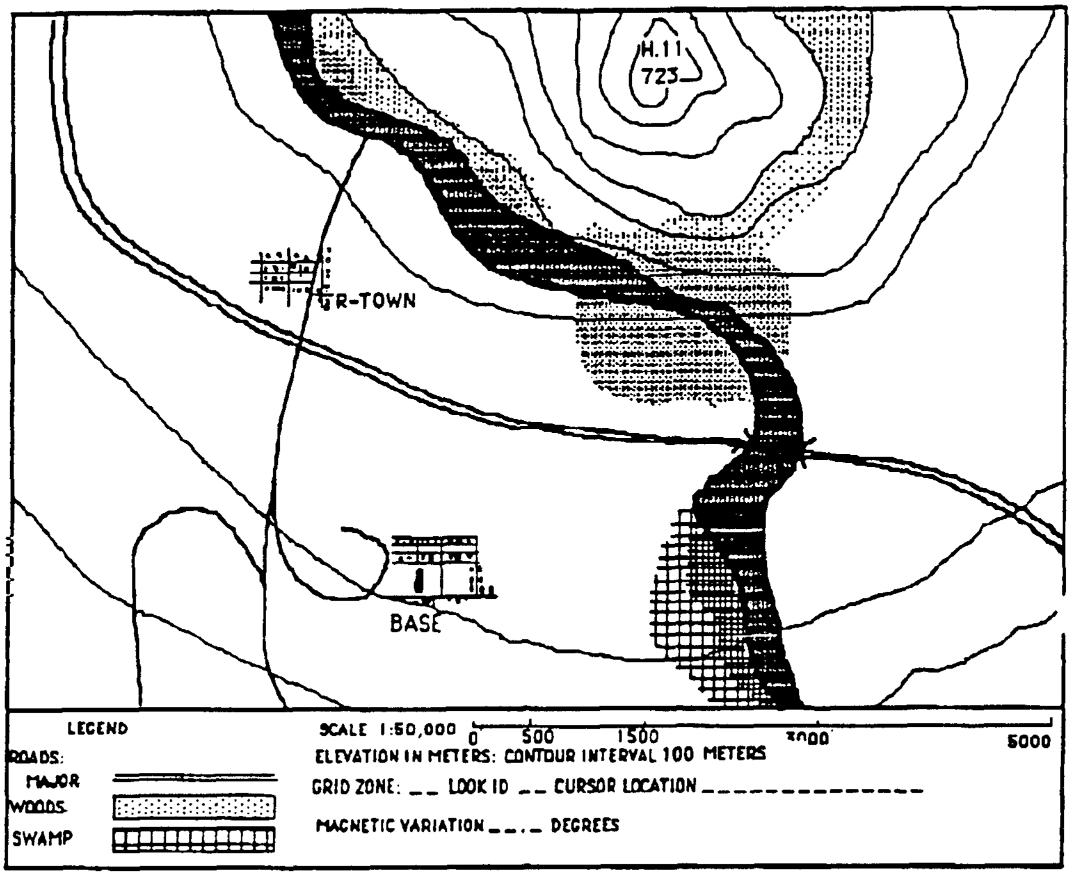

H.11
723
R-TOWN
BASE
LEGEND
ROADS:
MAJOR
WOODS
SWAMP
SCALE 1:50,000
0
500
1500
3000
5000
ELEVATION IN METERS: CONTOUR INTERVAL 100 METERS
GRID ZONE: __ LOOK ID __ CURSOR LOCATION ______________
MAGNETIC VARIATION __.__ DEGREES

Figure 10-1. Typical Electronic Map in Black and White

10.1.1.1 Curvature

Be consistent in projecting the earth's curvature on flat surface maps when displaying large geographic areas. Provide the user with a method of determining the type of map projection used.

10.1.1.2 Situation Display Presentation

Provide a means of presenting situation displays as overlays on related map backgrounds.

10.1.1.3 Map Label Position

Position map labels consistently (e.g., beneath or within the feature). Label all significant features without cluttering the display, where possible.

10.1.1.4 Map Orientation

Use a consistent map orientation when more than one map will be displayed (e.g., north consistent for all maps). It is recommended that all maps be north-oriented and the north direction annotated (see Figure 10-2).

10.1.1.5 Designating Map Areas

Consider using color, shading, texture patterns, or highlighting to define map areas of special interest. Shades (tones) of a single color are preferable to multiple colors when observers must make relative comparisons between or among areas. When using shades of color or texture patterns, the gradation of shades from dark to light should correspond to variation in the variable that is represented (see Paragraph 4.3.3).

10.1.1.6 Automated Tools

Provide automated tools for complex map analyses. The specific tools should be based on the user's needs. For example, avenue of approach, line-of-sight, and trafficability are needed by some but not all users. Determine user requirements, and provide appropriate tools.

10.1.1.7 Selectability

Enable the user to select a single item within a densely packed group. When a graphics item is selected, highlight it.

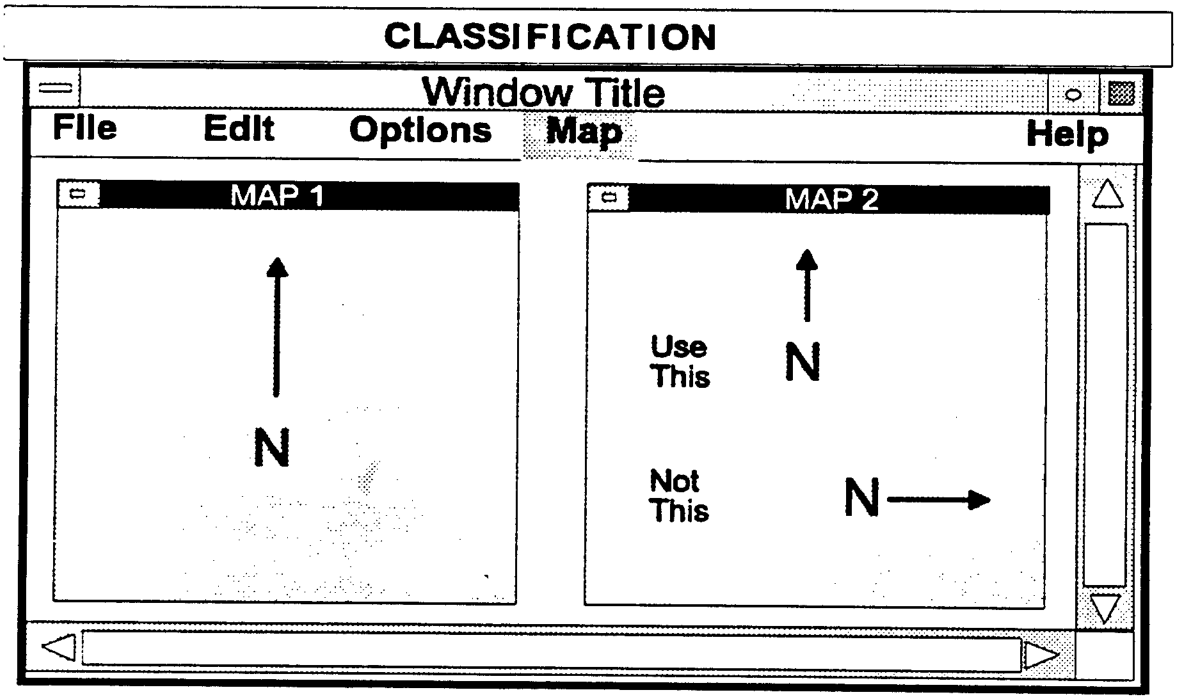


Figure 10-2. Example of Consistent Map Orientation

10.1.2 Static Display Attributes

10.1.2.1 Coverage Area and Resolution

As a minimum, ensure that maps cover user areas of responsibility at each organizational level, and provide all essential details required to conduct operations. Map displays should be large enough to permit the simultaneous presentation and visual integration of information required by the user. Small electronic displays may be panned and zoomed to increase map coverage. See Paragraph 10.1.3. However, at present, such displays have significant visual limitations when compared to traditional, large-format, paper maps.

- Ensure that all critical map features are represented.

- Ensure that labels remain legible at all display resolutions.

- Provide a means for reducing clutter while preserving essential information.

- Given a land-based command and control application, enable maneuver commanders at each echelon to view their own areas of operation, activities one echelon above and two echelons below, and activities of friendly adjacent (flanking) units. Also, display the activities of adjacent and deep enemy units that oppose displayed friendly forces (see Figure 10-3).

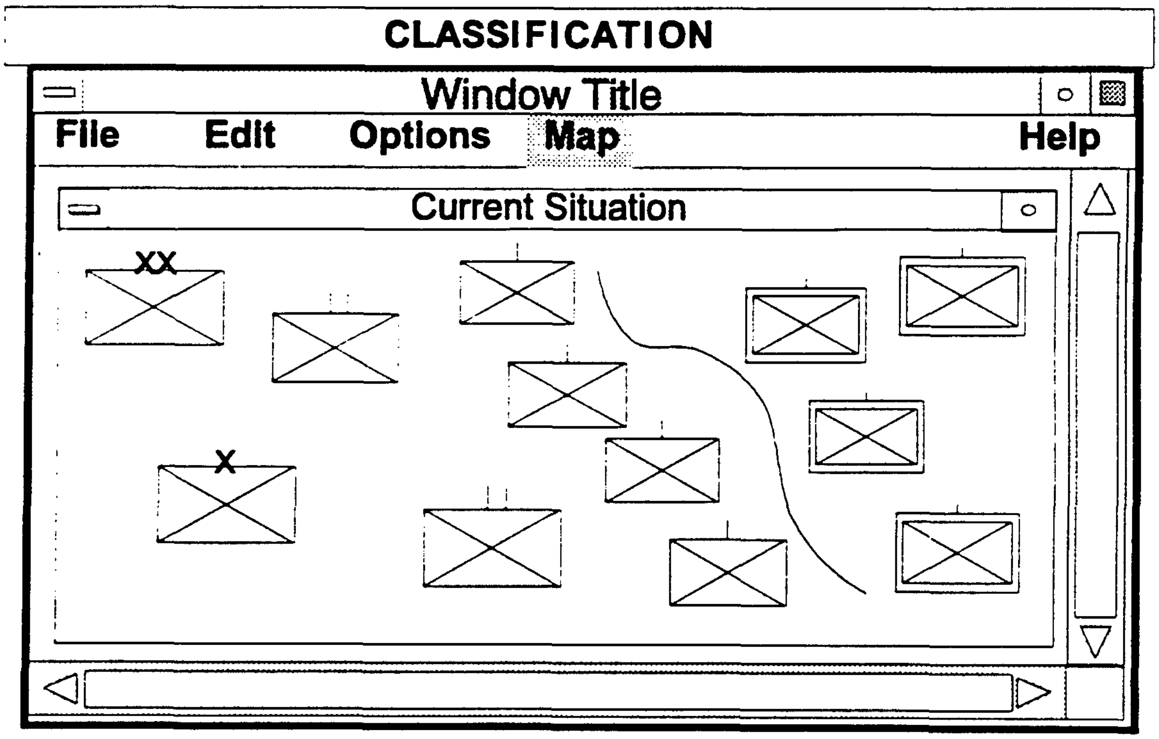


Figure 10-3. Brigade's Map Overlay Showing One Echelon Higher and Two Lower

10.1.2.2 Accuracy of Location

- Place symbols accurately on the map or connect to the desired location using arrows, lines, or other pointing graphics.

- Provide an automated means of registering graphic data with background map information at all display scales.

10.1.2.3 Symbology

Ensure that colors, symbols, line size/quality, and fonts are consistent throughout a given system. When possible, display symbology should conform to published standards (e.g., Army Field Manual 101-5-1 [1985b], NATO Standardization Agreement 2019 [1990], or the DIA Standard Military Graphics Symbols manual, 1990 draft), but each system should also be able to use a commercial graphics editor to accommodate the creation and display of system-unique features and symbols. The following guidelines are recommended:

- Use standard military symbols in accordance with doctrine when preparing maps and overlays. For example, the Army should use the current edition of FM 101-5-1, *Operational Terms and Symbols*.

- Provide a means by which the user may obtain help in identifying unknown symbols or other map information. For example, the user could highlight a symbol and query its meaning through a context-sensitive help feature.

- Use standard military map color codes, and provide a user-prompted key defining the color codes that are used (see Subsection 4.3).

- Do not allow map symbols to overlap, particularly if this would obscure their identity. Provide a means for moving background symbols to the foreground or otherwise revealing masked symbols where overlap is unavoidable.

- Display essential labels (e.g., unit identification) with the symbol; otherwise, provide a means by which the user can display information related to selected symbols. Figure 10-4 illustrates how a user could query a symbol for more detail.

- Consider the auxiliary use of alphanumeric coding where graphic data are not already so labeled.

- Position symbol labels consistently in accordance with doctrinal guidance.

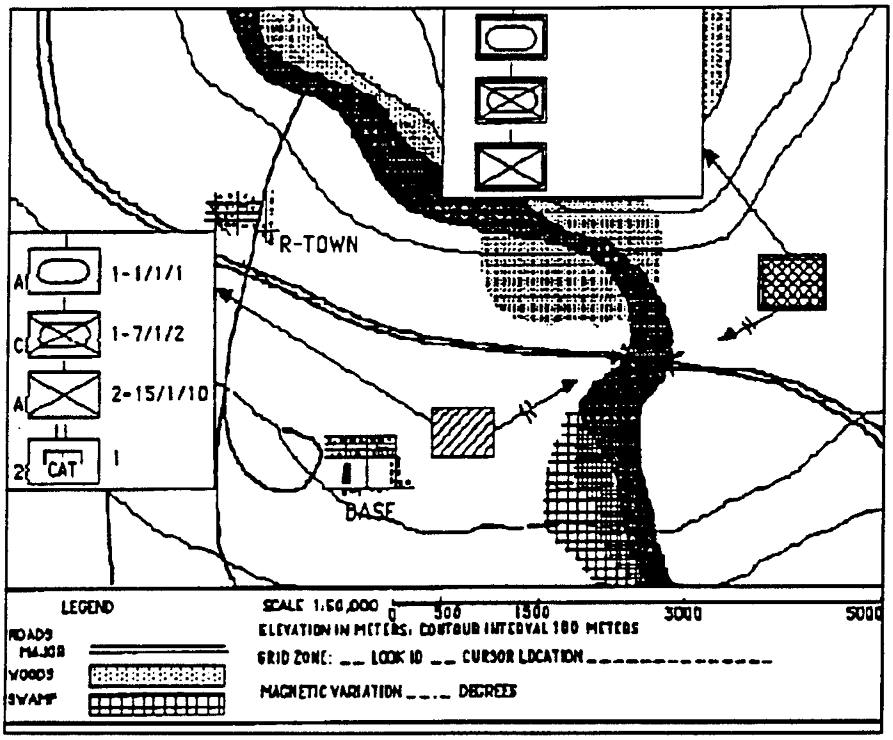

R-TOWN
A
1-1/1/1
C
1-7/1/2
A
2-15/1/10
2
CAT
1
BASE
LEGEND
ROADS
MAJOR
WOODS
SWAMP
SCALE 1:50,000
0
500
1500
3000
5000
ELEVATION IN METERS, CONTOUR INTERVAL 100 METERS
GRID ZONE: __ LOOK 10 __ CURSOR LOCATION______________
MAGNETIC VARIATION ____ DEGREES

Figure 10-4. Querying a Summary Symbol for Detailed Information

- Digital terrain and elevation data (DTED), available from Defense Mapping Agency (DMA) for some versions of electronic map (e-map), provide information that allows alternative methods of portraying terrain features. In addition to traditional topographic contour intervals, DTED can provide data for map overlays depicting road networks, drainage, vegetation, and soil type. Use shading, coloring, or other visual cues to accentuate terrain features.

10.1.2.4 Location of Displayed Section

Display a constantly visible display of coordinates associated with the cursor in user-selectable coordinate units, which can be changed conveniently where location information is often used. Augment continuous display of location with the capability to fix (point on map) a location to facilitate moving overlay displays. The coordinate display should be capable of displaying multiple coordinate units concurrently.

- Provide to the user a means of obtaining the exact map coordinates for a selected symbol or map feature by means of querying the symbol or feature. The recommended method of querying an item is to use a pointing device to place the cursor on the graphic to be queried and "click" the pointing device.

- When the entire map is not displayed, provide an inset that shows where the displayed portion is located within the larger map (see Figure 10-5).

- Provide an automated means for readily determining the distance between points.
- Provide a means for readily determining the bearing between points.

10.1.2.5 Area Bounding Boxes

Use bounding boxes when displaying maps in the main graphics drawing area. Area bounding boxes are pairs of coordinates defining a rectangular area, for example, latitude and longitude. Display the bounding coordinates for the geographic area being shown.

10.1.3 Dynamic Characteristics

In a map graphics application, make functions available through menus to permit the user to make measurements, perform analysis, and control the appearance of the display. A method is needed to scan and change the scales of the maps because of the limited screen size of many displays. In addition, changes in the tactical situation require updates to various map overlays. The following guidelines should be considered when implementing dynamically changing maps.

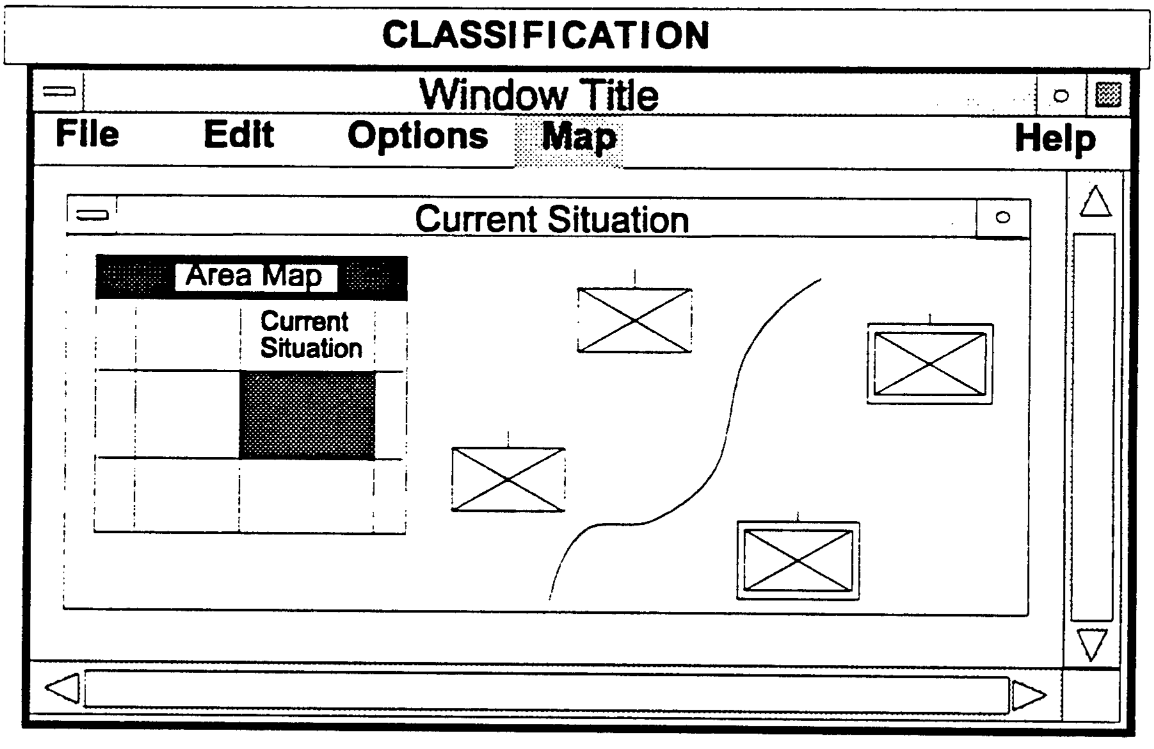


Figure 10-5. Example of a Map Inset

10.1.3.1 Panning

- Permit the user to change the displayed area by moving a window over the map in any direction. Panning operations may be continuous (preferable) or discrete but should meet the user's requirements.
- During panning operations, provide an indicator of position in the overall display.
- During panning operations, provide a means for rapidly returning to the starting point.

10.1.3.2 Zooming

- Provide a means for moving away from or toward the displayed area (zooming) to obtain a larger view or greater detail.
- Ensure that zooming does not cause problems in reading symbols, labels, or other map features.
- It is recommended that the level of detail (number of symbols and features depicted) be modified to match the degree of zooming used (i.e., more detail for close-up views and less for large-area perspectives).
- Of the two methods of zooming (i.e., continuous and discrete), continuous is preferable. Ensure that the method used is satisfactory to the user.
- When zooming, collapse symbols into fewer summary symbols to declutter.
- Provide a means for quickly returning to the normal display size when zooming.
- When changing scales through zooming, provide an indicator that continually shows the appropriate scale.
- It is recommended that an inset or window be provided that shows the maximum available map coverage. An example of map coverage (see Figure 10-5) would be a graphic square on the inset map that indicates the position of the map currently displayed. In the most useful form, this inset would be interactive and used to set parameters for calling up a screen map display.

10.1.3.3 Automatic Updating

Automatic updating, editing, and distributing map data are among the primary advantages offered by electronic displays. The following guidelines address considerations in implementing these capabilities:

- As appropriate, allow the user to select categories of information that will be automatically updated.

- Provide stable reference elements (e.g., terrain features, boundaries, etc.) when displays are automatically updated.

- Provide a means for readily identifying updates or changes. Critical changes must be easily recognized and distinguishable from other changes to the display. For example, highlight the update until the user acknowledges it.

- Allow the user to control how often the display is updated and to freeze the display to prevent further updates.

- Ensure that the rate of display update matches the perceptual abilities of the observer to permit successful visual integration of the changing patterns.

- Permit the user to freeze the display to prevent further updates. Provide a warning while the automatic display updating is suspended and when resuming automatic updating. Provide an option to either resume at the current time or at the time updating was suspended.

10.1.3.4 Sequencing

Display sequencing may be used to reduce clutter (e.g., presenting map overlays in succession), to reproduce temporal changes in the display database (e.g., changes in the tactical situation), and to aid in visualizing simulated changes in the battlefield situation.

- Allow the user to control the rate of sequencing where possible.

- Provide a capability to pause or suspend sequencing operations and provide an indicator of the status of sequencing operations.

- Allow the user to present sequenced displays in forward or reverse order as appropriate.

- Provide a means for the user to return quickly to a selected display within a sequence of displays.

- Consider using animation as an aid to the pictorial display for complex objects.

10.1.3.5 Grid Overlay

Provide a user-selectable grid overlay that is keyed to the coordinate system of the map. It should be easy for the user to turn the grid on and off. Coordinate keying of the overlays must be clearly specified and easily operated by the user.

10.1.3.6 Dynamic Map Legend

Ensure that the map display has an associated window giving relevant information in a continuous display. The information should include map scale, cursor location, graphic of map coverage, and status (i.e., "working," "computing," "available," etc.).

10.1.3.7 Cursor Design

Ensure that the cursor includes a point designation feature (e.g., cross hairs or a v-shaped symbol), because fine accuracy is often required in positioning the cursor.

10.1.3.8 Distance/Azimuth

Provide a distance/azimuth function that calculates the distance (range) and azimuth (bearing) between any two selectable points or symbols. Present distance in selectable units (feet, meters, miles, or kilometers). Azimuth should be displayed in degrees from true north.

10.1.3.9 Position Determination

The "determine position" function calculates the position of the point that is identified, and the answer should be presented in a selectable coordinate system (e.g., Universal Transverse Mercator, latitude/longitude, or Military Grid Reference System). It is recommended that answers be provided textually in user-specified units of measure, such as latitude and longitude, distance (in nautical miles), and azimuth (in degrees from true north).

10.1.4 Creating And Editing Map Graphics

10.1.4.1 Standard Symbol Library

Provide a library of standard symbols and a means of copying and manipulating symbols.

10.1.4.2 Labeling Symbols

Provide an easy means of labeling symbols. Consider automated means of aiding the user in labeling and enforcing labeling conventions.

10.1.4.3 Building Symbols and Overlays

Provide automated tools to assist the user in constructing new symbols and graphics overlays.

10.1.4.4 Printing Preview

When preparing graphics displays for printing, allow users to preview displays as they will appear when printed.

10.1.4.5 Display Editing

- Allow the user to add or delete symbols, labels, or other features without destroying background information.
- Allow the user to expand an area of the display as required for accurate placement of critical data.

- Provide a means for designating graphic elements for editing. Highlight selected items to provide a visual cue of forthcoming subsequent actions.

- Allow the user to reposition selected elements on the display.

- Allow the user to remove and restore selected elements.

- Allow the user to select from displays of available options when making changes to display attributes (e.g., color, symbols, line types, textures, etc.). Selection should be made by pointing rather than by naming the options.

- Provide an easy means for the user to identify attributes currently selected.

- Provide the user an easy means to change the attributes of selected graphic elements.

- Provide an easy means for naming, storing, and retrieving graphics displays and elements. Also, provide a means for reviewing and selecting from stored graphics files.

10.1.5 Map Display Characteristics

10.1.5.1 Map as a Base Screen

When an application is map intensive, it is recommended that the map be used as the background or base screen, which should be the maximum display size possible to promote readability.

10.1.5.2 Map Readability

Ensure the readability of map features, since the map is the focus of the user. When possible, the screen design should avoid displays that cover the map, and windows should not obscure the map. .

10.1.5.3 Map Cursors

For map cursors, use a cross-hair design that has high contrast with the background. It is recommended that cursor size subtend 20 minutes of visual angle so the average user can easily locate it on the map.

10.1.5.4 Graphic Overlays

An overlay is a layer of information (e.g., grids, boundaries, control measures) that has been drawn on a graphics canvas. Make various overlays available to the user to display (make visible), hide from display (make invisible), or delete. The preselection or filtering of graphic overlays is a recommended feature. The decluttering of graphic displays (especially maps) should be assisted.

- Carefully review labels and titles used to identify filters to ensure items are understandable. The filters should be extended to map features, such as roads, cities, vegetation, topography, and political data. The feature overlay displays should use standard map symbols as a default (e.g., railroads, dams, and roads). The intensity of the map should be controllable to allow fadeout of the map without losing all the map features.

- Graphic overlays may overlap map features but should not obscure text information. The text may be offset with arrows to preserve map legibility.
- Include in the graphics package the capability to display a list of available overlays, distinguishing between visible and invisible overlays.
- Other possible overlays include boundary lines, oceans, rivers, grids, air fields, railways, and user-generated overlays (created through the graphics editor).
- Provide a map overlay editor function.
- Prevent the creation of overlays with the same name or title. Display the overlay feature legends at user request.

10.1.5.5 Color Use with Graphic Overlays

Using color to identify symbols is encouraged, but also ensure that it is redundant with another type of coding. This caution is especially true for friend-enemy or danger-safe designations. Dots, dashes, shapes, and video effects are recommended. Be careful to avoid visual color illusions caused by color blending (e.g., adjacent red and blue lines are seen as one purple line).

10.2 PRESENTATION GRAPHICS (GRAPHS, PICTURES, AND DIAGRAMS)

Graphs should be used where necessary to visualize relationships among two or more variables, to facilitate comparing sets of data, to aid the observer in visualizing trends in data, and to aid in extrapolating future values of the underlying data. Graphs are also useful when comparing actual data to predicted values, when comparing actual values to established limits in control processes, for representing rapidly changing data, and for interpolating values between known points. In general, graphs have advantages over tabular data in summarizing complex relationships among variables and facilitate information processing and understanding.

Pictures are becoming an increasingly important form of graphic presentation. The multimedia capabilities of developing computer systems have increased the availability of pictures within computer applications. The most frequent operational picture is a map. The use of scanned maps has transferred the operational planning focus to the computer interface. The use of pictures on computer screens must be done with great care to avoid misleading the user.

Use diagrams (schematics) when the user requires information concerning the spatial relationship among objects but does not require the level of detail required by pictures. Schematic representations can be used as an aid to understanding relationships among components of complex systems and as a means of conveying status information concerning the operation of systems and their components. They also provide a medium through which users may manipulate designs and observe subsequent actions on modeled systems.

10.2.1 General

10.2.1.1 Complex Formats

Avoid complex formats, such as 3-D presentations and artistic embellishments (pictures, shading, colors, decorative items), which detract from the intended purpose of the graphic.

10.2.1.2 Clarity Preservation

Design graphics to preserve clarity when the graphics must be reproduced or reduced in size. Application window sizing should be controlled so no graphic shows partial lines.

10.2.1.3 Appropriateness of Formats

Provide formats (presentation styles) appropriate for the user's level of training and experience. Graphics should utilize user-expected symbols.

10.2.1.4 Data Specific to Task

Provide only those data the user needs for a specific task.

10.2.1.5 Alternative Style Selection

Allow users a selection of alternative presentation styles.

10.2.1.6 Querying Data Elements

Provide a means by which the user can select data elements on the graph and display the associated values.

10.2.1.7 Graphical Versus Tabular

Consider allowing the user to select between graphical and tabular data formats.

10.2.1.8 Consistency

Be consistent in design, format, labels, etc. for each presentation style.

10.2.1.9 Labeling

Clearly label the displayed graphics.

10.2.2 Creating And Editing

10.2.2.1 Computer-Aided Entry

Provide computer aids for the entry and organization of complex graphic data.

10.2.2.2 Data Entry Validation

Validate data entries. Automated validation may include comparison to a standard range and/or the use of rules for relationships among variables. The validation process should be part of the application software.

10.2.2.3 Data Entry Aids to Plotting

When plotting formats are known, provide templates or other data entry aids to facilitate the entry of graphic data.

10.2.2.4 Automated Plotting of Stored Data

Automate plotting of stored data, and provide the user with automated editing and construction capabilities.

10.2.2.5 Automated Production of Scales

Automate the production of scales, and/or provide the user with automated aids for scaling graphic data.

10.2.2.6 Lines

- Provide automated aids for drawing straight and curvilinear line segments.
- Use rubberbanding (i.e., provide a visible line that connects a starting point to current cursor position), which can be made permanent when selected.
- Provide automated assistance in joining and intersecting line segments.
- Allow the user to identify and select line segments for moving and editing. Typically, this is done through highlighting and dragging the line. This capability should include grouping of individual segments to allow actions to be taken on the grouped object.
- Provide optional, adjustable, grid references to aid the user in aligning horizontal and vertical lines.

10.2.2.7 Rule Specification by the User

Allow the user to specify rules for attributes, relationships, and design, and have the computer apply those rules automatically during the design process. For example, straighten hand-drawn lines, adjust angles between intersecting lines, and complete details of graphic elements.

10.2.2.8 Computer-Aided Drawing

Provide computer-aided methods for drawing figures and a system of prompts or other means to aid the user in the design process.

10.2.2.9 Automatic Scale Reduction

Allow the user to edit or create drawings using a large scale, which will later automatically reduce to the desired scale.

10.2.2.10 Object Manipulation

Provide a basic set of capabilities to resize, copy, move, and rotate displayed objects. Extend these capabilities to grouped objects.

10.2.2.11 Mirror Imaging

Provide a means of producing mirror images (reflecting) as an aid in producing symmetrical graphic displays.

10.2.2.12 Grouping Elements

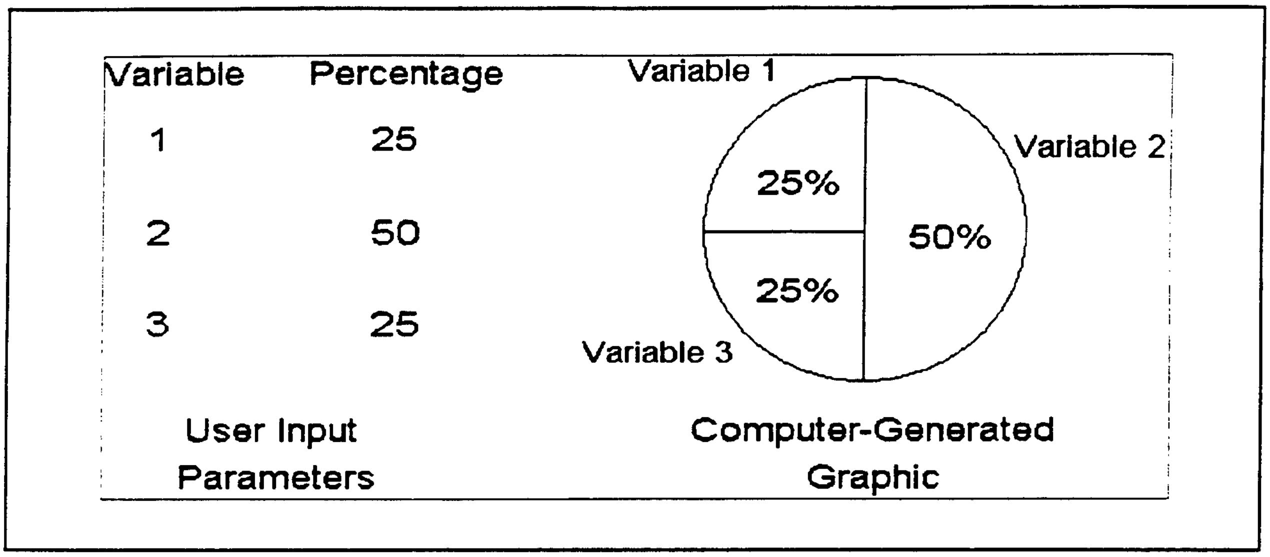
Permit the user to select and group graphic elements that will be edited in common.

10.2.2.13 Area Fill Capability

Provide an automatic means of filling enclosed areas with selected attributes (e.g., color or texture).

10.2.2.14 Computer Models for Graphical Display Generation

Provide computer models that can generate graphical displays in response to parameters provided by the user (see Figure 10-6).



Variable	Percentage
1	25
2	50
3	25

Figure 10-6. Example of How a Computer Model Can Generate Graphics From User Input

10.2.3 Scales, Labels, and Coding

10.2.3.1 Standard Scaling Conventions

Use standard scaling conventions: values on the horizontal axis increase to the right of the origin; values on the vertical axis increase going up from the origin. Independent variables (time or causal events) are plotted against the horizontal axis; dependent variables (effects) are plotted against the vertical axis.

10.2.3.2 Standard Meanings

Use or assign standard meanings to graphic symbols, and apply them consistently.

10.2.3.3 Color and Pattern Coding

Users prefer colors to patterns for coding lines or filling areas of graphs on visual displays. Good design requires redundant coding be used. See Subsection 4.3 for color usage guidelines. Use texture for coding on printed outputs, since in most cases color will not be available.

10.2.3.4 Texturing Displays

If texturing must be used, use simple hatching or shading, and avoid patterns that produce visual illusions of vibration and motion (see Figure 10-7).

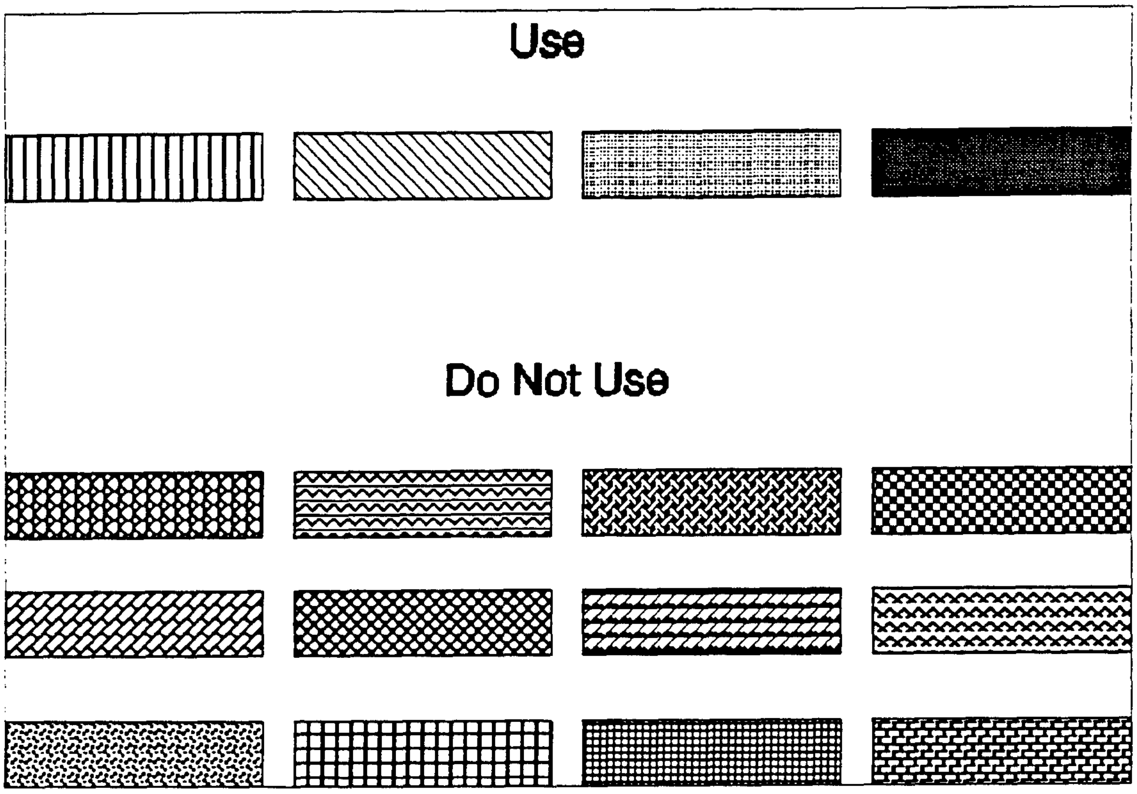


Figure 10-7. Texture Patterns

10.2.3.5 Axes Breaks in Expanded Scales

When expanding scales to emphasize a limited range of data, provide breaks in the axes to indicate discontinuities with the origin (see Figure 10-8).

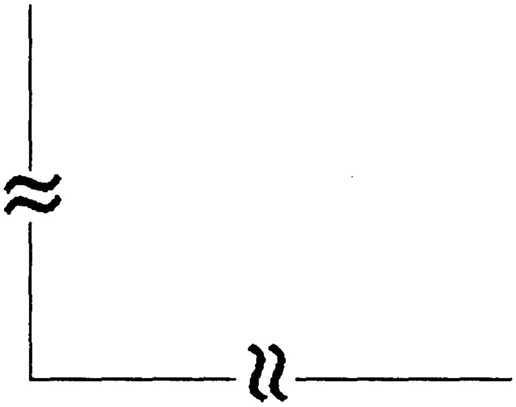


Figure 10-8. Example of Breaks in a Graph's Axes When Scales Have Been Expanded

10.2.3.6 Duplicating Axes

When scaled data contain extreme values, it may be difficult for the user to comprehend the scale values in relation to the data. To aid readability, add a copy of the X-axis at the top and a copy of the Y-axis at the right of the graph. Extreme values and data are thus in proximity throughout the graph. In some cases where numbers of extremely large and extremely small orders of magnitude populate the graph, a logarithmic scale may be necessary.

10.2.3.7 Avoid Exaggerated Scales

Avoid the use of exaggerated scales that distort or suppress trends in the data (see Figure 10-9).

10.2.3.8 Formats for Graphic Comparison

Provide identical formats and scales when comparisons are required between separate graphs, or plot different sets of data on the same graph.

10.2.3.9 Using Linear Scales

Linear scales should be used in preference to nonlinear scales whenever practical. For example, see Figure 10-10.

10.2.3.10 Using Logarithmic Scales

Logarithmic scales may be used where comparisons of rates of change and percentages are required or where numbers of both extremely large and extremely small orders of magnitude populate the graph.

10.2.3.11 Multiple Entries

Avoid multiple scales on the axes of a single graph.

10.2.3.12 Labeling Tick Marks

Number or label tick marks corresponding to major scale divisions on the axes, and include a label containing descriptions and units of measurement on each axis.

10.2.3.13 Numbering Scale Divisions

Where practical, use no more than 10 to 12 major scale divisions separated by up to 9 subdivisions. When the appearance of the display will not be degraded, major scale divisions should be decimal multiples of whole numbers, cover the entire range of the data, and start from zero.

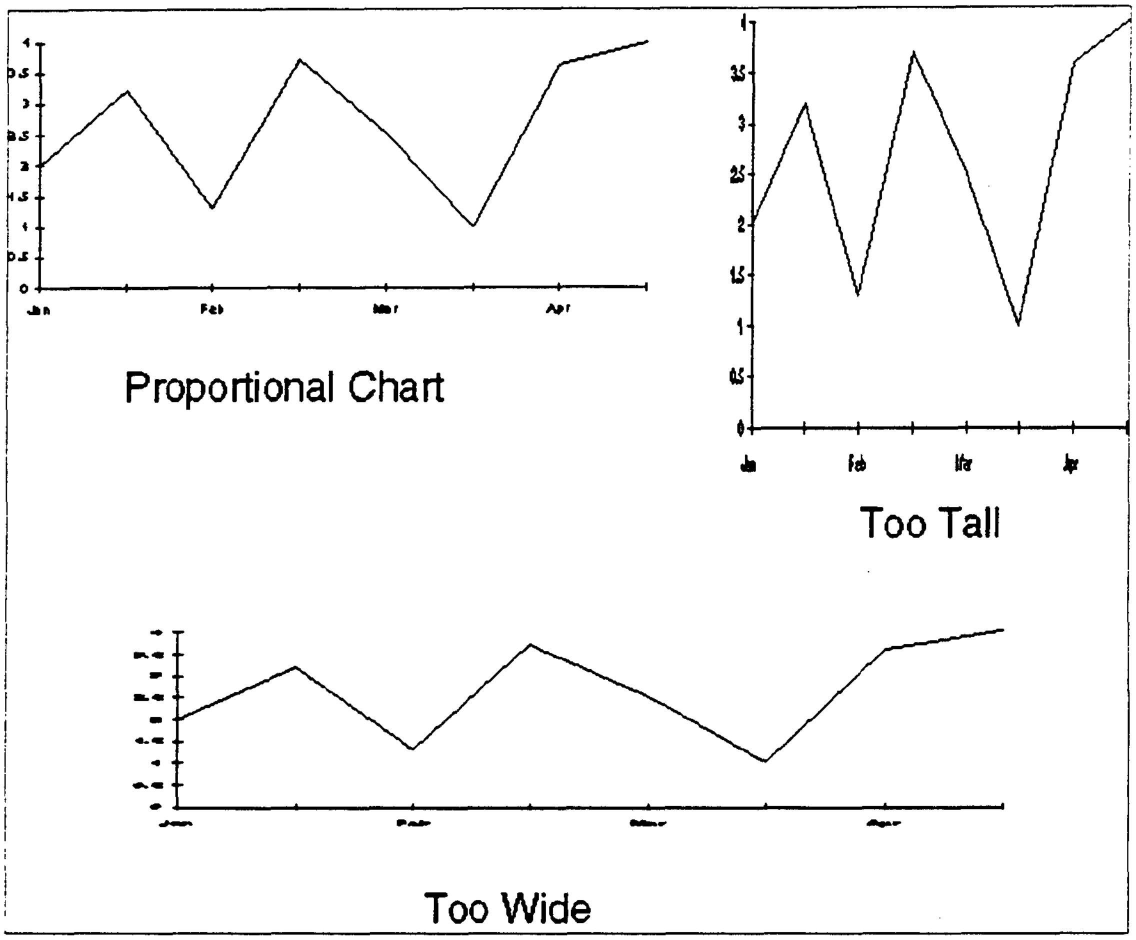


Figure 10-9. Comparing Distorted Data Trends Induced by Exaggerated Scales to a Proportional Scale

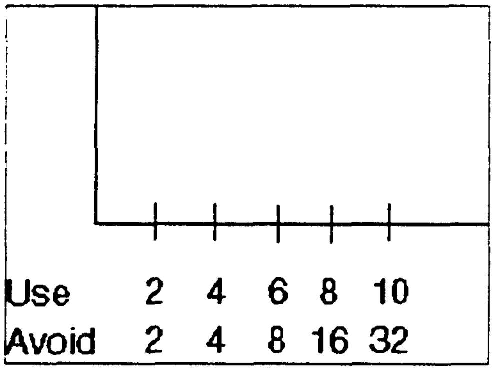


Figure 10-10. Example of Linear Versus Nonlinear Scales

10.2.3.14 Numeric Scale Division

Begin numeric data scales with zero when users must use displays to compare quantities or different series.

10.2.3.15 Label Format

Labels should use upper and lower case sans serif fonts and be oriented left-to-right for normal reading.

10.2.3.16 Use of Labels

Use labels in preference to legends or keys when it is necessary to identify plotted data elements. Orient labels horizontally and locate them adjacent to the referenced elements. Arrows, lines, or similar pointing conventions may also be used to connect labels to their respective data elements.

10.2.3.17 Location of Legends and Keys

Locate legends or keys that identify graphic data elements within the rectangular bounds of the graph, unless such positioning would interfere with interpretation of the displayed data.

10.2.4 Identifying Critical Data

10.2.4.1 Displaying Values

Display reference or baseline values when users are required to make comparative evaluations against a fixed standard.

10.2.4.2 Using Supplementary Text

Consider using supplementary text to emphasize features of data requiring user attention. See the example in Figure 10-11.

10.2.4.3 Displaying Data Values with Graphics

Display actual data values in addition to the graphic display where precise readings of values are required, as illustrated in Figure 10-11.

10.2.4.4 Position of Text Used for Labeling

When labeling graphic data, position text consistently with respect to graphic elements.

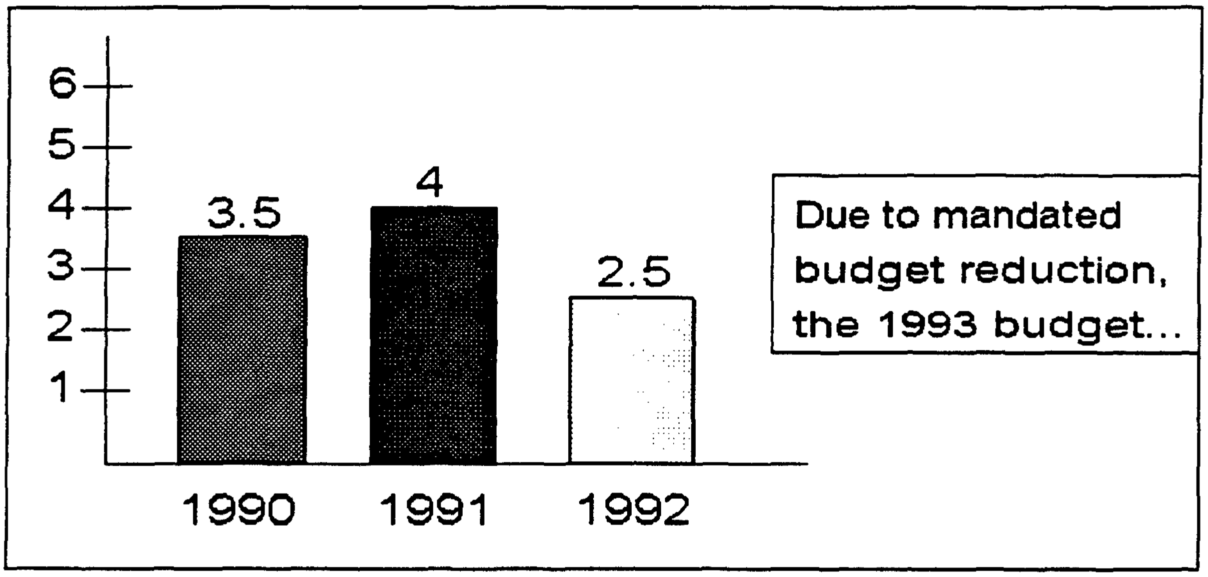


Figure 10-11. Use of Supplementary Text and Actual Values in a Graph

10.2.5 Grid Lines

A grid is the set of horizontal and vertical lines, including the labeled and scaled axes, which form a rectangular boundary around the graph. Additional horizontal and vertical grid lines corresponding to scale values partition the bounded area of the graph and provide a visual aid in locating and reading points on the graph(s). Use a grid and grid lines, as appropriate, when presenting data graphically.

10.2.5.1 Grid Line Visibility

Ensure that grid lines are easily distinguishable and do not obscure graphed data.

10.2.5.2 Using Grid Lines

Avoid excessive use of grid lines. Locate grid lines using the guidelines for placement of major scale values. Consider using more grid lines where greater precision is required or where the size of the display will permit their use.

10.2.5.3 User Display of Grid Lines

Where practical, allow the user to determine whether or not grid lines will be displayed.

10.2.6 Types Of Presentation Graphics

10.2.6.1 Curve and Line Graphs

- Use smoothed curves or straight lines connecting data points (line graphs) when displaying relationships between two continuous variables (e.g., when showing time variation in some quantity).
- When a single graph contains multiple curves, designate each curve with an adjacent label. If it is necessary to use a legend, list legend codes in the order in which curves occur in the graphs.
- When displaying multiple curves, highlight a curve containing critical data.
- Use line coding to distinguish among multiple curves on the same graph, and use coding consistently when the same types of data appear on different displays.
- Use a distinct line code (e.g., dashed or dotted lines) when projecting values beyond the actual data set.
- For cyclic data, provide at least one full cycle of data.
- Consider plotting the difference between two series where comparisons are necessary.

10.2.6.2 Area Charts

Area charts provide a means of visualizing the relative contributions of individual elements to the sum of their individual parts, often as a function of time. Figure 10-12 uses an area chart to illustrate how the total number of items (i.e., the sum of the numbers within each category) varies with time.

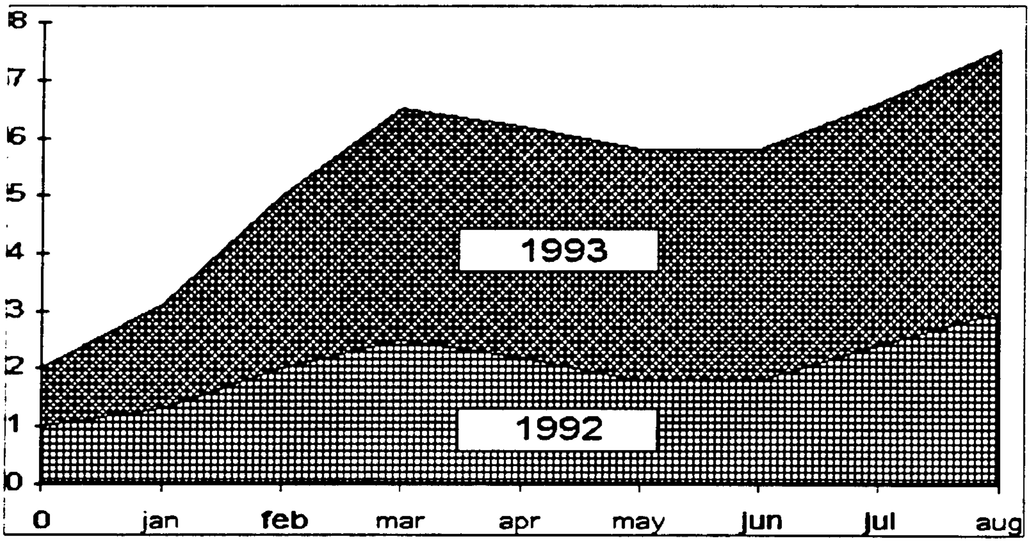


Figure 10-12. Example of an Area Chart

- Use texture or shading to indicate the area between curves.
- Stack the series with the least variable series at the bottom and the most variable at the top.
- Place labels within the textured or shaded bands if space is available.

10.2.6.3 Bar Graphs

Bar graphs represent the magnitudes of numeric data by the lengths of parallel bars. Bars may be vertically or horizontally oriented and are usually spaced apart along an axis containing discrete reference points (e.g., months, mid-points of sample intervals, non-numeric categories, etc.). Histograms, or stepcharts, are bar graphs without spacing between bars, used when a large number of intervals must be plotted. Figure 10-13 illustrates bar graphs. Graphic presentations should be designed to conform to user expectations.

10.2.6.4 Scatterplots

Scatterplots present data as a 2-D distribution of points and should be considered when necessary to show how variables are related or to represent the spatial distribution of data (e.g., impacts on a target). Highlight particularly significant data points. Figure 10-14 illustrates a scatterplot.

10.2.6.5 Pie Charts

Pie charts, like bar graphs, are used to show proportional distribution of categories with respect to sum of the categories. See example in Figure 10-15.

- Place labels in a normal orientation on the segments of pie charts. Segment labels should include numbers that indicate percentages and/or absolute numbers represented by each segment of the display.
- Segments requiring emphasis should be highlighted or displaced slightly from the rest of the pie chart, as illustrated in Figure 10-15.

10.2.7 Pictures

10.2.7.1 Using Pictures

Consider using graphic pictures when a very detailed representation of objects is required. For example, see the scanned map in Figure 10-16.

10.2.7.2 Automated Aids for Pictures

Provide automated aids when users must perform detailed analyses of image data.

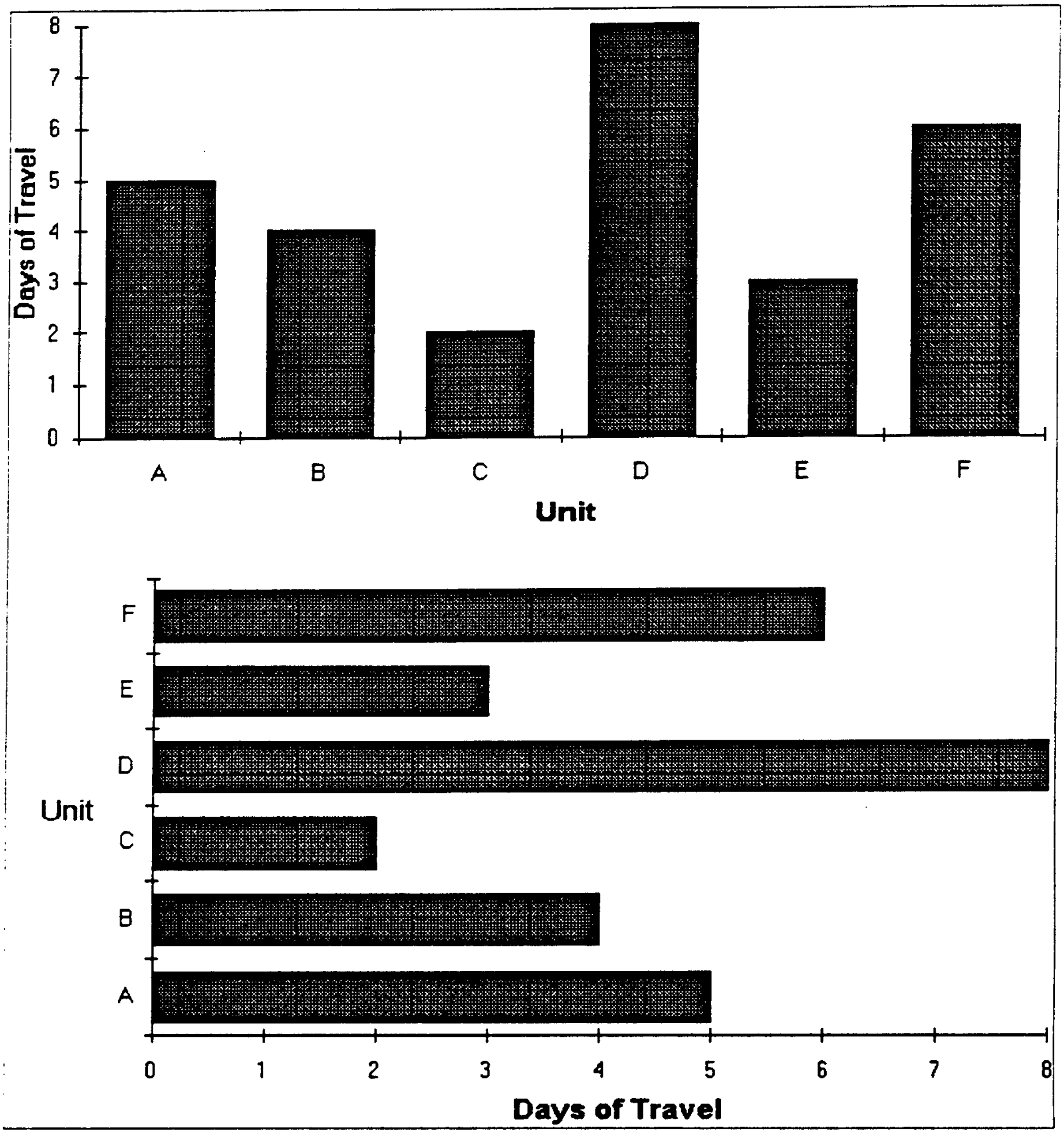


Figure 10-13. Examples of Bar Graphs

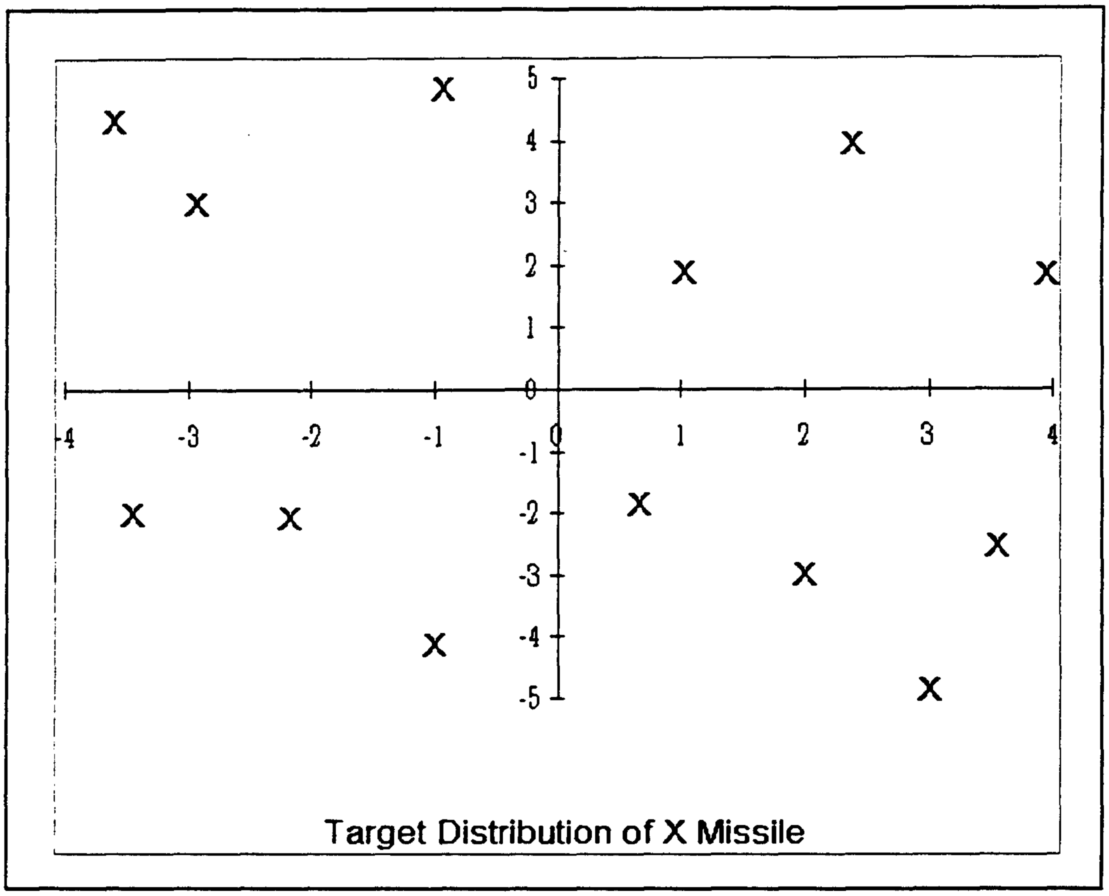


Figure 10-14. Example of a Scatterplot

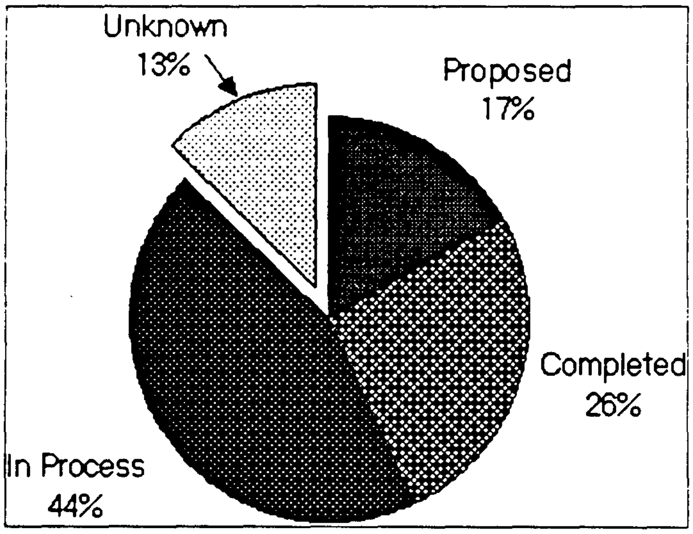


Figure 10-15. Example of a Pie Chart

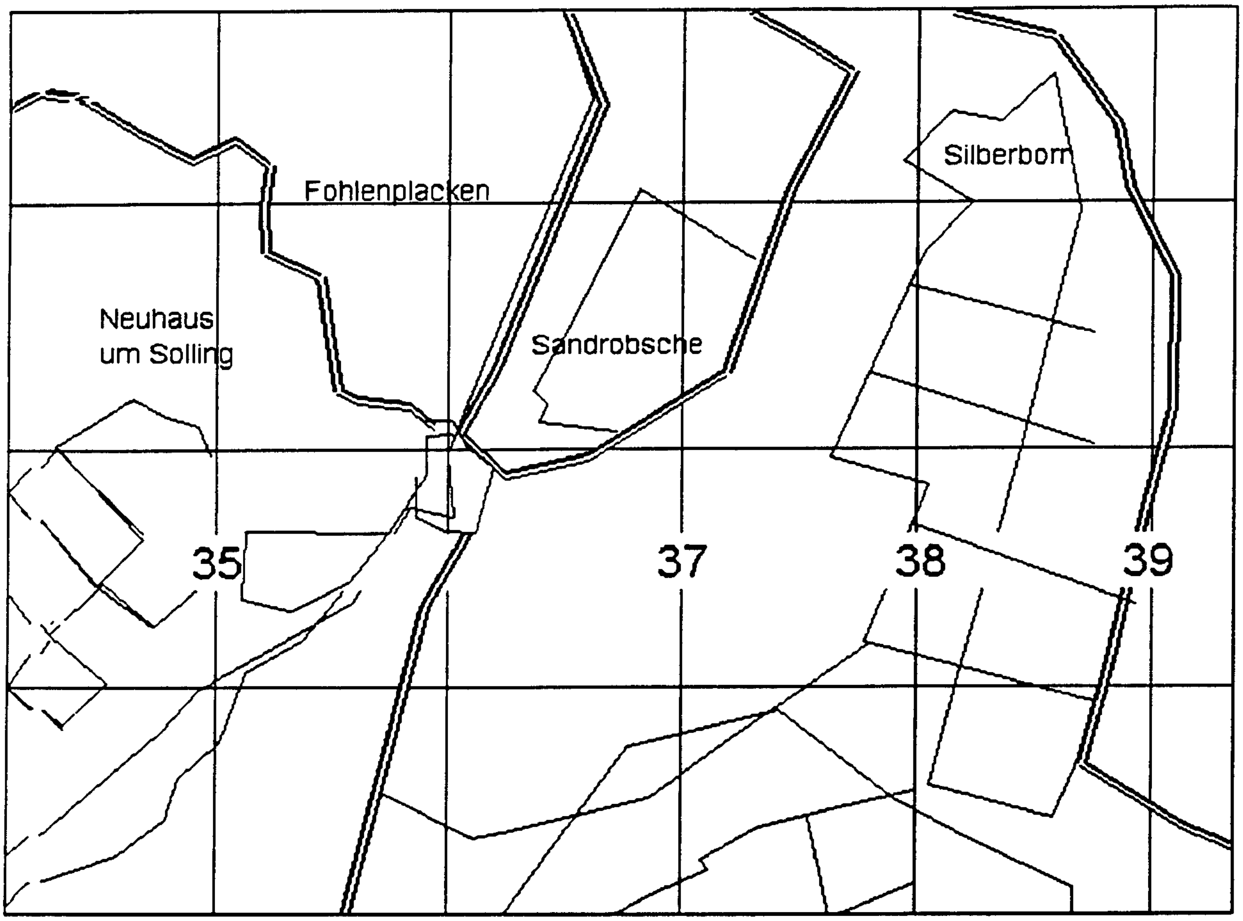


Figure 10-16. Example of a Graphic Picture

10.2.8 Diagrams (Schematics)

10.2.8.1 Diagrams General

Use diagrams when user requires information concerning the spatial relationship among objects but does not require the level of detail provided by pictures.

- When diagrammed data are presented in separate sections, use consistent notations across sections, provide an easy means for users to move among sections, and provide an overview of the entire diagram represented by the individual sections.

- Highlight portions of diagrams requiring special user attention.

- Provide a capability for the user to rotate displayed diagrams where it is necessary to view the object from different perspectives.

10.2.8.2 Flowcharts

Use flowcharts to provide a schematic representation of sequential processes. Use them also as aids to problem-solving when solutions can be reached by answering a series of questions. Figure 10-17 illustrates a typical flowchart.

- As appropriate, sequence flowchart elements in a logical order; otherwise, when designing flowcharts, minimize path lengths to reduce size.
- The layout of flowchart paths should conform to standard orientation conventions (i.e., left to right, top to bottom, or clockwise).
- Consistently apply the coding schemes for flowchart elements.
- Use standard directional conventions when using arrows to connect elements of flowcharts.
- Use highlighting to direct a user's attention to elements of particular significance.
- When using flowcharts as decision aids, require only one decision at each step, and provide the user with a logically ordered list of available options.
- Use consistent wording for options displayed at decision points.

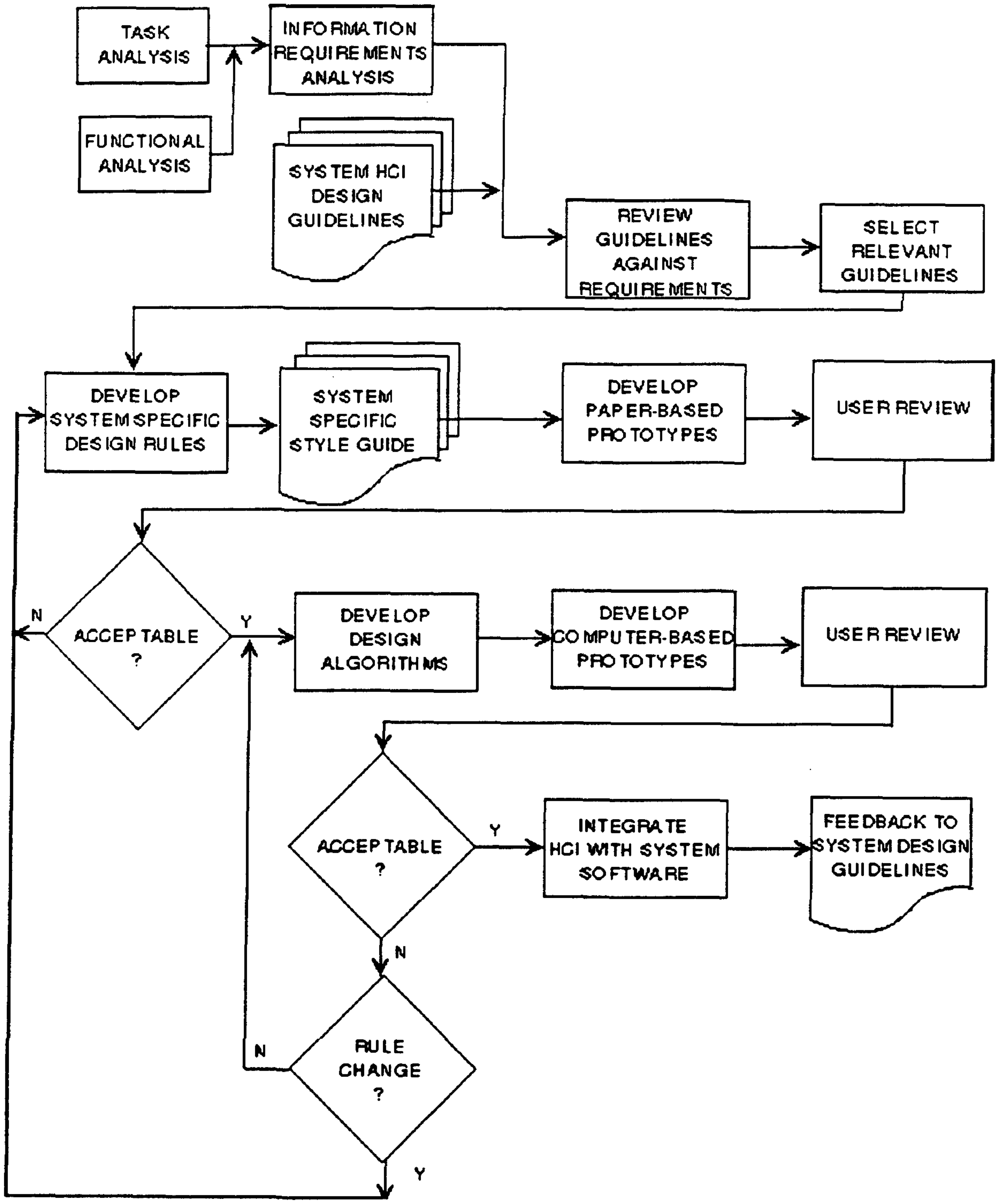

TASK ANALYSIS
FUNCTIONAL ANALYSIS
INFORMATION REQUIREMENTS ANALYSIS
SYSTEM HCI DESIGN GUIDELINES
REVIEW GUIDELINES AGAINST REQUIREMENTS
SELECT RELEVANT GUIDELINES
DEVELOP SYSTEM SPECIFIC DESIGN RULES
SYSTEM SPECIFIC STYLE GUIDE
DEVELOP PAPER-BASED PROTOTYPES
USER REVIEW
ACCEPTABLE ?
N
Y
DEVELOP DESIGN ALGORITHMS
DEVELOP COMPUTER-BASED PROTOTYPES
USER REVIEW
ACCEPTABLE ?
Y
INTEGRATE HCI WITH SYSTEM SOFTWARE
FEEDBACK TO SYSTEM DESIGN GUIDELINES
N
RULE CHANGE ?
N
Y

Figure 10-17. Sample Flowchart

REFERENCES

Paragraphs	References
10.1.1.1	Smith and Mosier (1986) para 2.4.8-3
10.1.1.2	Smith and Mosier (1986) para 2.4.8-3
10.1.1.3	Smith and Mosier (1986) para 2.4.8-15
10.1.1.4	Smith and Mosier (1986) para 2.4.8-4 and 2.4.8-5
10.1.1.5	Smith and Mosier (1986) para 2.4.8-7 through 2.4.8-9
10.1.1.6	Smith and Mosier (1986) para 2.4.8-13
10.1.2.2a	Lewis and Fallesen (1989) para 2.5.6.1.1.2
10.1.2.2b	Smith and Mosier (1986) para 1.6-17
10.1.2.3	U.S. Department of the Army (1985b)
10.1.2.3a	Lewis and Fallesen (1989) para 2.5.6.1.1.1; U.S. Department of the Army (1985b)
10.1.2.3d	Lewis and Fallesen (1989) para 2.5.6.1.1.3
10.1.2.3e	Lewis and Fallesen (1989) para 2.5.6.1.1.4
10.1.2.3f	Smith and Mosier (1986) para 2.6-8
10.1.2.3g	Lewis and Fallesen (1989) para 2.5.6.1.1.4.6; U.S. Department of the Army (1985b)
10.1.2.3h	Lewis and Fallesen (1989) para 2.5.6.1.3
10.1.2.4	Bowser (1991) p. 13
10.1.2.4a	Lewis and Fallesen (1989) para 2.5.6.1.1.2 and 2.5.6.2.5
10.1.2.4c	Smith and Mosier (1986) para 2.4.8-12
10.1.3	Bowser (1991) p. 13
10.1.3.1a	Lewis and Fallesen (1989) para 2.5.6.2.3
10.1.3.1b	Smith and Mosier (1986) para 2.4.8-11
10.1.3.2e	Lewis and Fallesen (1989) para 2.5.6.2.4
10.1.3.2f	Smith and Mosier (1986) para 2.4-16
10.1.3.2g	Smith and Mosier (1986) para 2.4-17
10.1.3.3b	Smith and Mosier (1986) para 2.4.8-18

REFERENCES (cont'd)

Paragraphs	References
10.1.3.3e	Smith and Mosier (1986) para 2.7.3-4
10.1.3.4e	Smith and Mosier (1986) para 2.4-18
10.1.3.5	Bowser (1991) p. 14
10.1.3.6	Bowser (1991) p. 14
10.1.4.2	Bowser (1991) p. 15
10.1.4.3	Smith and Mosier (1986) para 2.4-20
10.1.4.5b	Smith and Mosier (1986) para 1.6-5
10.1.4.5c	Smith and Mosier (1986) para 1.6-6 and 7
10.1.4.5e	Smith and Mosier (1986) para 1.6-9
10.1.4.5f	Smith and Mosier (1986) para 1.6.10 and 11
10.1.4.5g	Smith and Mosier (1986) para 1.6-12
10.1.4.5h	Smith and Mosier (1986) para 1.6-13
10.1.4.5i	Smith and Mosier (1986) para 1.6-15 and 16
10.1.5.1	Bowser (1991) p. 14
10.1.5.2	Bowser (1991) p. 14
10.1.5.3	HFS (1988); Bowser (1991) p. 14
10.1.5.4	Bowser (1991) p. 14
10.1.5.4b	Bowser (1991) p. 15
10.1.5.5	Bowser (1991) p. 15
10.2	Smith and Mosier (1986) para 2.4.6-2, 5.2
10.2.1	Lewis and Fallesen (1989) para 2.1.1-3; Brown (1989) para 5.3-8, 5.10; Brown (1989) p. 88
10.2.1.1	Lewis and Fallesen (1989) para 2.3.6
10.2.1.2	Bowser (1991) p. 15; Lewis and Fallesen (1989) para 2.3.9
10.2.1.4	Smith and Mosier (1986) para 2.4-5
10.2.1.6	Lewis and Fallesen (1989) para 2.3.2.2b
10.2.1.7	Lewis and Fallesen (1989) para 2.3.2.2c

REFERENCES (cont'd)

Paragraphs	References
10.2.1.8	Smith and Mosier (1986) para 2.4-4
10.2.2.1	Smith and Mosier (1986) paras 1.6.1-1 and 1.6.1-18
10.2.2.2	Smith and Mosier (1986) para 1.6-19
10.2.2.3	Smith and Mosier (1986) para 1.6.1-2
10.2.2.4	Smith and Mosier (1986) paras 1.6.1-2 and 1.6.1-4
10.2.2.5	Smith and Mosier (1986) paras 1.6.1-5 and 1.6.1-6
10.2.2.6a	Smith and Mosier (1986) para 1.6.2-1
10.2.2.6b	Smith and Mosier (1986) para 1.6.2-2
10.2.2.6c	Smith and Mosier (1986) para 1.6.2-3
10.2.2.6d	Smith and Mosier (1986) paras 1.6.2-4 and 1.6.2-5
10.2.2.7	Smith and Mosier (1986) paras 1.6.2-6, 1.6.2-7 and 1.6.2-18
10.2.2.8	Smith and Mosier (1986) paras 1.6.2-8 and 1.6.2-9
10.2.2.9	Smith and Mosier (1986) para 1.6.2-11
10.2.2.10	Smith and Mosier (1986) paras 1.6.2-10, 1.6.2-12 and 1.6.2-13
10.2.2.11	Smith and Mosier (1986) para 1.6.2-14
10.2.2.12	Smith and Mosier (1986) para 1.6.2-15
10.2.2.13	Smith and Mosier (1986) para 1.6.2-17
10.2.2.14	Smith and Mosier (1986) para 1.6.2-19
10.2.3.1	Smith and Mosier (1986) para 2.4.1-1
10.2.3.2	Smith and Mosier (1986) para 2.4-12
10.2.3.3	Brown (1989) para 4.2.7
10.2.3.4	Smith and Mosier (1986) para 2.4-14; Lewis and Fallesen (1989) para 2.4.7.1
10.2.3.5	Smith and Mosier (1986) para 2.4.1-7
10.2.3.6	Smith and Mosier (1986) para 2.4.18
10.2.3.9	Lewis and Fallesen (1989) para 2.3.1.3
10.2.3.10	Lewis and Fallesen (1989) para 2.3.1.4

REFERENCES (cont'd)

Paragraphs	References
10.2.3.11	Lewis and Fallesen (1989) para 2.3.1.5
10.2.3.12	DoD (1989a) para 5.15.3.6.4
10.2.3.13	Lewis and Fallesen (1989) para 2.3.1.8-9
10.2.3.14	Smith and Mosier (1986) para 2.4.1-7
10.2.3.15	Lewis and Fallesen (1989) para 2.3.3
10.2.3.16	Lewis and Fallesen (1989) para 2..4-5
10.2.3.17	Lewis and Fallesen (1989) para 2.3.5
10.2.4.1	Smith and Mosier (1986) para 2.46
10.2.4.2	Smith (1986) para 2.4-7
10.2.4.3	Smith and Mosier (1986) para 2.4-8
10.2.4.4	Smith and Mosier (1986) para 2.4-9
10.2.5	Lewis and Fallesen (1989) para 2.3.2
10.2.5.1	Lewis and Fallesen (1989) para 2.3.2.1
10.2.5.2	Lewis and Fallesen (1989) para 2.3.2.1.1
10.2.5.3	Lewis and Fallesen (1989) para 2.3.2.2a
10.2.6.1a	Smith and Mosier (1986) para 2.4.3-1
10.2.6.1b	Smith and Mosier (1986) para 2.4.3-3, 2.4.3-4
10.2.6.1c	Smith and Mosier (1986) para 2.4.3-5
10.2.6.1d	Smith and Mosier (1986) paras 2.4.3-6 and 2.4.3-7
10.2.6.1e	Smith and Mosier (1986) para 2.4.3-8
10.2.6.1f	Smith and Mosier (1986) para 2.4.3-10
10.2.6.1g	Smith and Mosier (1986) para 2.4.3-11
10.2.6.2a	Smith and Mosier (1986) para 2.4.3-12
10.2.6.2b	Smith and Mosier (1986) para 2.4.3-13
10.2.6.2c	Smith and Mosier (1986) para 2.4.3-14
10.2.6.3	Bowser (1991) p. 15; Smith and Mosier (1986) para 2.4.4-3

REFERENCES (cont'd)

Paragraphs	References
10.2.6.4	Smith and Mosier (1986) para 2.4.2-3
10.2.6.5	Smith and Mosier (1986) para 2.4.5-1
10.2.6.5a	Smith and Mosier (1986) paras 2.4.5-2 and 2.4.5-3
10.2.6.5b	Smith and Mosier (1986) para 2.4.5-5
10.2.7	Smith and Mosier (1986) para 2.4.6-1
10.2.7.1	Smith and Mosier (1986) para 2.4.6-6
10.2.8.1a	Smith and Mosier (1986) para 2.4.6-3
10.2.8.1b	Smith and Mosier (1986) para 2.4.6-3
10.2.8.1c	Smith and Mosier (1986) para 2.4.6-5
10.2.8.2	Smith and Mosier (1986) paras 2.4.7-1 and 2.4.7-2
10.2.8.2a	Smith and Mosier (1986) paras 2.4.7-3 and 2.4.7-4
10.2.8.2b	Smith and Mosier (1986) para 2.4.7.5
10.2.8.2c	Smith and Mosier (1986) para 2.4.7-6
10.2.8.2d	Bowser (1991) p. 16; Smith and Mosier (1986) para 2.4.7-7
10.2.8.2e	Smith and Mosier (1986) para 2.4.7-8
10.2.8.2f	Smith and Mosier (1986) paras 2.4.7-9 and 2.4.7-10
10.2.8.2g	Smith and Mosier (1986) para 2.4.7-12

This page intentionally left blank.

11.0 DECISION AIDS

DoD continues to develop automated decision aids in support of user tasks. Although what constitutes a decision aid has been debated, it is important to point out that decision aids assist, rather than replace, human decision-makers. Consequently, when defining decision aids, applications limited to managing information are usually excluded, as are those that make decisions in a fully autonomous mode.

The question of "when to use decision aids" is reviewed in the first part of this section. Given that decision aids are to be used, the next step is to define the requirements. When the requirements are firm, the features needed to support the requirements become important. Section 11.0 then deals with specific issues of decision aid interface design.

Decision aids may be designed to be parts of other software or as stand-alone applications. For example, decision aids have been designed to assist users in evaluating military courses of action. These applications present alternatives and supporting evidence, as well as assist the user in evaluating the alternatives. The user retains a major role in developing the final recommendations. Information management software such as database management, text processing, and graphics applications may support the decision process but are not usually considered decision aids. Other applications, such as engine diagnostic software, may include many relatively fixed rules derived from human experts. Such "expert systems" can include many properties of decision aids, but they place relatively more emphasis on internal rules to arrive at conclusions. Examples of autonomous systems include automatic fire control systems and robotic devices. These systems may require human supervision but rely heavily on internal rules and algorithms for their operation.

It is difficult to make a distinction between decision aids and expert systems (which include autonomous capabilities), since both require cooperation between human and automated system components. Holtzman (1989) differentiates between expert and intelligent decision systems and points out which is appropriate to use based on subject matter, circumstance, and preference of the decision-maker. Expert systems may have a relatively large knowledge base and rules that respond to constant environmental factors, whereas decision aids place more burden on the decision-maker. Decision aids provide assistance and are designed to help in uncertain or novel situations. The guidelines presented in this section are appropriate for both expert systems and decision aids. Therefore, the term "decision aid" or "aid" will be used to refer to both types of decision support applications.

11.1 USE OF DECISION AIDS

Decision aids should be used to compensate for known limitations in human decision-making and to offset the adverse effects of external factors. In general, difficulties can arise because of the fundamental limits of human cognitive (i.e., mental) abilities and lack of experience. Difficulties also arise because of various environmental factors that both stress the decision-maker and determine the type, quantity, quality, and rate of information presented.

11.1.1 Cognitive Considerations

Consider the cognitive limitations and styles of decision-makers when designing decision aids. For example, overload (i.e., stress, information, situational, etc.) often causes users to focus on a subset of the available information. Innate abilities and learned information-processing strategies may cause additional problems. The following points describe several commonly occurring limitations.

- **Cognitive Limitations** - Novices, individuals lacking confidence, and those performing tasks under stressful conditions will make errors that may result in less-than-optimal decisions. For example:
  - Humans often have difficulty retrieving, retaining, representing, and manipulating large amounts of information. They also may have difficulty combining multiple cues or criteria or performing computational tasks. These difficulties result in delaying performance or avoiding difficult tasks.
  - Humans often have difficulty making decisions in times of uncertainty.
  - Novices do not have previous experience, so they often fail to recognize errors.
  - When making a decision, humans often simplify decision problems by selectively perceiving data, information, and knowledge. They may set outcome objectives and then look for a decision that meets them. Thus, they may focus on confirming rather than on refuting evidence. They also may adjust decision methods to fit goals or desired results.
  - If a decision leads to a negative result, humans may attribute the outcome to chance or to the complexity of the problem, rather than to their own decision-making deficiencies.
  - Humans have limited memory available for current tasks and will lose some information within seconds.
  - Humans have limited abilities to organize information.
  - Humans usually have difficulties with symbolic and quantitative manipulation of mental representations, and they may have difficulty formulating or dealing with abstractions.
  - Humans may have difficulty extrapolating time and space information.
  - Humans may fail to use prior experience to generalize in new situations.
- **Pitfalls of Complexity** - Humans often have difficulty dealing with complexity and may, therefore, try to make a problem less complex by avoiding certain aspects of it. Consequently, they may not consider all factors when making decisions. Some strategies humans use to make decisions less complex are:

- Humans often simplify decision problems by only considering a few alternatives.
- Humans may use only part of the available information. Or they may use information that corresponds to a mental representation or model of what they imagine the solution to be, even if this means rejecting or misperceiving relevant information. They may combine or "chunk" information in various ways, rely on poor memory-search strategies, or rely on erroneously perceived correlations between data.

- **Cognitive Biases** - Humans have biases that can carry over into the decision-making process. Humans may:
  - Recall information that has been recently acquired, frequently rehearsed, or semantically related to current information
  - Anchor their judgments (i.e., place greater emphasis on early evidence) and then fail to adjust when provided new information
  - Give preference to information they believe is causally related to the problem
  - Provide numeric judgments that contain systematic bias or variance
  - Select cues that are often unreliable indicators of the true situation
  - Use inappropriate analogies to generate and compare options
  - Incorrectly identify current situations with similar past events
  - Fail to detect unique features among similar cases, and inconsistent or ambiguous information may not be noticed or emphasized appropriately.
- **Time Allocation** - Humans may fail to allocate time properly to different phases of the planning process. Too much attention to early stages of planning may leave inadequate time to evaluate derived alternatives properly. Humans may:
  - Perform detailed analysis early but fail to do so later
  - Fail to develop and evaluate the alternatives thoroughly
  - Fail to identify, evaluate, compare, and combine salient information and, therefore, fail to identify, prioritize, and assess goals
  - Fail to model (war-game) alternatives because of lack of time.

11.1.2 External Factors

Many external factors may influence the quality of decisions.

- **Information overload** - When the complexity, dynamics, and/or volume of information to assess are high (such as in battlefield operations), they may degrade decision-making performance.

- **Time stress** - Humans have difficulty analyzing information fast enough to meet external time constraints. When they have enough time, they often have difficulty maintaining high performance long enough to analyze all data.

- **Limited information** - Decision-makers may work in situations where information available to support their decisions is limited. Under such circumstances, lack of experience and human limitations in making or formulating estimates may pose problems.

- **Training** - Decision-makers may not have the experience, deductive skills, or knowledge of the procedures necessary to make a decision. A decision aid can assist by performing some steps, by leading the human through the required steps, and by filling in knowledge gaps. Properly designed decision aids also train users through explanation and embedded training.

11.1.3 When to Use Decision Aids

Use decision aids to help the user overcome the difficulties previously described. The following are examples.

11.1.3.1 Manage Complexity

Decision aids help the user cope with information overload; they focus attention. Use decision aids when the user is trying to manipulate large amounts of data or visual representations, combining multiple criteria, allocating resources, managing detailed information, and selecting and deciding among alternatives.

11.1.3.2 Improve Timeliness

A decision aid helps a user perform many time-consuming activities more quickly. Some examples include diagnosing the current state of a system and mathematical calculations, particularly when they are beyond the user's abilities. Providing aid when users encounter unfamiliar problems also helps improve the timeliness of the process.

11.1.3.3 Best Use of Limited Data

Use decision aids when limited data result in uncertainty. Decision aids help by predicting future events from limited information, improving the accuracy and reliability of critical tasks, and addressing critical areas beyond the ability of the user.

11.1.3.4 Overcoming Limitations

Use decision aids to overcome the human cognitive limitations described in the earlier sections. For example:

- Use decision aids to overcome human limitations in dealing with uncertainty.
- Decision aids are helpful in overcoming emotional components of decision-making.
- If the quality of human performance is in question, decision aids can add greater accuracy to the process.
- Decision aids are ideal in cases where memory- and information-retention problems exist.
- Decision aids overcome cognitive biases well.

11.1.4 When to Consider Alternatives

The following are circumstances under which the use of decision aids may not be advisable.

11.1.4.1 Obvious Solutions

Do not use decision aids when solutions are obvious or when one alternative clearly dominates all other options.

11.1.4.2 Time Requirements

Use decision aids only when sufficient time is available or when the user is authorized to make decisions.

11.1.4.3 Generalizing

As appropriate, defer to the human ability to generalize.

11.1.4.4 Adaptation

Recognize situations where individuals may be superior in adapting to novel situations.

11.1.5 Cautions and Limitations

Exercise caution when introducing decision aids, in particular, when they include functions that reduce the role of human judgment.

11.1.5.1 User Complacency

The decision aid may encourage users to take a less active role. This, in turn, may cause users to be inattentive and less prepared to handle sudden decreases or increases in workload, both of which may reduce accuracy.

11.1.5.2 Continued Vigilance

If the user's role becomes less active, the user may have difficulty maintaining sustained attention, which may lead to longer user response times.

11.1.5.3 Discrimination Limitations

It is necessary to recognize limitations in a user's ability to discriminate between correct and incorrect automated decisions.

11.1.5.4 Fear of Automation

Many users mistrust automation and automated decisions, preferring to believe their performance is superior to that of automated systems. User attitudes toward automation are often based on the fear of being replaced. The designer should take this into account when planning the role and degree of authority the user will have in overriding automated decisions.

11.2 DEFINING DECISION AID REQUIREMENTS

Develop decision aids or expert systems that focus on tasks that users find difficult, rather than on what is already done routinely.

11.2.1 Understand Tasks

Base designs on an in-depth understanding of both the tasks to be performed and the conditions of their performance.

- The best way to define decision aid requirements is to start with experts in the field. However, it is important to choose the experts appropriately. Be sure to use more than one expert and verify that they really are experts. If they have knowledge of part of the field, be sure to consider those parts, and find other experts for the other parts. Also, ensure that common users participate with the group of experts. When obtaining information from the experts, provide a means to identify the criteria used to reach decisions.

- Decision aids must be matched to the situation and limitations they are designed to support.

- Recognize that not all functions are appropriate for decision aids. Determine the appropriate functions and design them to be compatible with the user's decision processes.

- Provide no more than one aid for each task.

11.2.2 Understand Requirements

Decision aid development should be driven by requirements, not by technology.

- Identify areas where users actually need help, then match the decision aid to the needs of the intended users.
- Recognize the user's decision situation and goals, and focus on the highest-level goal.
- Anticipate skepticism concerning automated decision support. Recognize that the dominant factor in accepting decision aids is perceived utility. The system must add new capabilities or increase efficiency in the performance of decision-making tasks.
- Consider characteristics of the user population in designing the decision aid and its interface.

11.2.3 Types of Aids

Types of aids and presentation formats may vary according to the phases of the decision process (i.e., alerting, acquisition, evaluation, and responding) and factors such as time stress.

11.2.4 Function Allocation Between Humans and Computers

Allocating functions between humans and computers must be based on cognitive task analysis, not on what is achievable using current technology.

- Recognize that aided performance may not exceed unaided performance, even though aided methods are preferred.
- Decision aids and expert systems can enhance decision quality. However, they may increase the user's workload because users may be required to consider more variables. Seek design alternatives that prevent or minimize increased workload.
- The aid must be complete for its intended purpose. Address all critical aspects of the decision situation.
- Recognize that a user's decision-making behavior is contingent upon the task and context within which it is performed. Design the decision aid to provide decision methods suitable to probable variations in tasks and context.
- Users often prefer to perform some of the tasks and allow the decision aids to perform others. Specifically, users prefer to do the easy to moderately difficult tasks and leave difficult tasks to the decision aid. This interaction is necessary to maintain user interest and attention. Decision aids are more acceptable to users if viewed as advisors rather than decision-makers.
- Avoid applications that are trivial or lack complexity because they may undermine the value of automated decision support methods.

11.3 FEATURES OF DECISION AIDS

11.3.1 General Design Considerations

- Ensure that a decision aid is easier to use than the decision process it replaces. It must be flexible, versatile, and easy enough to benefit typical users (i.e., users don't need to be subject matter experts). A decision aid must use terminology and criteria appropriate to the target user group. It must be easy to control and understand.

- Ensure that a decision aid is capable of responding to the user's ad hoc requests in time to allow the information to influence decisions. The interface should facilitate the exchange of information.

- Tailor decision aids to the resources available to the user.

- Ensure that a decision aid automatically identifies meaningful patterns and relationships and brings them to the attention of the user.

11.3.2 Provide Decision Alternatives

- Ensure that a decision aid is able to support development and evaluation of multiple, feasible alternatives. The aid should present a set of possible alternatives, each of which could be feasible. However, the aid should not display all of the options when that would be too complex. The decision aid also should display which goals are served by the different alternatives and applicable options.

- Ensure that the decision aid supports user evaluation of decision options. First, the aid should generate alternatives for the user to evaluate and should allow the user to input his or her own alternative(s). Second, the aid should have a method of assigning and explaining probabilities for alternatives. The user should be able to explore different solutions, including using different decision strategies and criteria. Once the user has applied all desired options, the aid should rank-order the decision alternatives. This assistance also should include guidance in using rating procedures.

11.3.3 Prediction, Simulation, and Modeling

- Ensure that the application is able to predict future data. Historical data should be available to make comparisons, search for precedents, and assist the user in visualizing trends. The decision aid should alert the user when it predicts a future problem or opportunity upon which the user needs to act.

- Provide a modeling and simulation capability to support "what if?" exercises and to make predictions based on current conditions.

- Ensure that models used in decision aiding are appropriate, designed to answer specific questions, and validated.

11.3.4 Identify and Assess Factors Underlying Decisions

- Provide a means of obtaining and assessing weights for multiple criteria. Multiple criteria should be statistically independent, when possible. As appropriate, provide a means of combining weights from multiple sources. This refers to the technique of multi-attribute decision-making.

- Identify and rank causal factors by their importance, and assign weights. The application should allow users to modify the decision factors and their weights and to provide and adjust risk factors used in decision models. This refers to techniques such as pair-wise comparison.

- Ensure that the aid is able to explain the contributions of underlying factors and supports the use of sensitivity analysis for exploring those contributions. The aid must identify and assess operational constraints and provide a means of informing the user (upon request) of decision aid boundaries or other limitations. The aid should make available to the user the assumptions underlying modes and parameters and a history of the aid's past performance.

- Ensure that the decision aid makes it easy for the user to provide input into the aid's decision. The user should be able to add new decision factors and set the range of conditions (within the decision aid's set limits), the level of output detail, and the parameters for optimization. Provide a means for saving and reusing the user's modifications, but also provide a means to return to the default settings.

- Assist in visualizing interacting factors.

- Provide a means for assuring the validity of elements added to the decision model, in particular those used over successive applications.

11.3.5 Handling Decision Aid Recommendations

- Ensure that the application is able to calculate and display results of selected decision options.

- Ensure that the application provides facilities for assessing costs, risks, and benefits of all alternatives.

- For users to trust the decision aid, the aid must explain the rationale behind outputs or recommendations. The aid also should provide indicators of certainty or uncertainty when making recommendations.

- When data are missing or uncertain, ensure that the aid identifies this situation and gives information on the possible impact on the recommendations.

- Ensure that the decision aid includes internal consistency checks to prevent the system from making contradictory predictions and recommendations.

- Ensure that the decision aid informs the user when it cannot handle the current situation.

11.4 USER REQUIREMENTS

11.4.1 General Considerations

- Ensure that the decision aid is user-friendly, beneficial to the user, and presents information that is readily understood by or familiar to the users. Where possible and appropriate, ensure that the decision aid has sufficient "intelligence" to adjust to user task requirements.

- Ensure that the decision aid uses decision methods acceptable to the decision-maker and is able to accommodate user changes. Once the decision method is determined, the user must retain control throughout the process. The aid should provide feedback on the method and the current stage of processing.

- Reduce the user's data-entry requirements as much as possible. To do so, set defaults for data-entry fields. However, these defaults and fields must be user-changeable.

- Ensure that a decision aid automatically alerts users to important new developments occurring in the database or as a result of predictive modeling.

- Ensure that the system encourages the user to participate in the decision process. To do this, the system should represent problems and solutions in the same way the users do. The system also should try to foster user "ownership" of decisions and allow the user to exercise judgment over the decision aid results. This includes providing sufficient information to the user both about the process and about the end result.

- Ensure that the decision aid guides the user through the process, providing automated guidance on how to define and analyze a problem and formulate a decision. When user input is required, the decision aid should help make this requirement clear. However, it should not make the user dependent, such that the process cannot be completed when the system is unavailable.

- Avoid presenting too much data. Use aids to reduce, filter, and preprocess data into a form useful to the decision-maker.

- Avoid increasing the user's work load, when possible. Prepare users for changes and possible increases in work effort when necessary, and point out the aid's abilities to increase effectiveness.

- Reduce complexity. A major reason for using decision aids is to simplify the user's task. Therefore, some guidelines are necessary on the amount of information to present to the user. In general, the system should provide information required to perform the tasks allocated to the user; however, it should only present information relevant to the task being performed. The system should provide no more information than is essential and should avoid repeating already available information. Present the information using a level of abstraction, resolution, or detail appropriate to the immediate task.

- When time is limited, ensure that the system anticipates the user's needs and provides a greater degree of autonomous decision-making.

- Ensure that users are able to extend and personalize the decision aid. However, provide a means to validate models created or modified by users, and provide sufficient warnings about the consequences of failures to validate. Ensure that the decision aid can be easily returned to a default state.

- Ensure that the decision aid analysis is flexible to the user's needs and desires; give the user control over the data retrieval and analysis process. The user should select the degree of analysis to be done and time frame to be considered. When the system asks questions, the user should have the option of either changing the question or not answering. The system should also accommodate the various information requirements of commanders and staff users, including the ability to adjust the level of detail. It should be able to create a user profile containing preferences and jargon.

- Provide procedures appropriate to the user's level of expertise. Designers should recognize that experts may use mental imagery; novices depend more on rule-based procedures.

11.4.2 Decision Aid Interface

- Ensure that the user-machine interface supports an intelligent dialogue between the user and the decision aid. For example, it should adapt to the user; understand the user's goals, needs, and abilities; interpret poorly formulated queries; correct user errors; and overcome user limitations. The interface also should reflect the tasks to be performed and should be tailored to the resources available.

- Ensure that the system helps prevent the user from making errors. When errors are made, it should provide automatic error recovery.

- Apply user-interface design guidelines mentioned elsewhere in this document.

- Ensure that the decision aid allows users to customize formats to their own needs. However, it is preferable to minimize the user's requirements to make such changes. To do this, the application should associate and group data in a meaningful way, and displays should match the task.

See Subsection 8.2 of this *Style Guide* for detailed information on HELP applications.

11.4.3 Explanations

Decision aids should be capable of providing domain-specific explanations to answer user questions and must be capable of guiding the user through the decision process, as well as providing procedural help on system use.

- When the system provides explanations, ensure they are easy for the user to understand. Explanations should use terms familiar to the user, incorporating the user's concept of the problem and maintaining consistency with the immediate task. Intuitive explanations or analogies are helpful for topics that are likely to be too difficult for the user to understand.

- Length of explanation is important. Provide a short explanation initially, with the ability to provide more detail at the user's request. Consider how much to tell the user. Weigh trade-offs in what the user can learn about the decision aid and what the decision aid can/should explain to the user.

- Assist the user in locating key elements of the decision model, as related to a specific decision task.

- Provide the capability to explain the current decision model or method, and be prepared to justify the use of component factors. Document the decision aid's algorithms, and make them available for user inspection.

11.4.4 Training

- Provide backup systems and appropriate training in performing any user tasks replaced by decision aids. When decision aids are available, provide regular training to the user in all skills required to maintain proficiency on backup systems. This training will be necessary if decision aids become unavailable. Training may be preferable to using decision aids for handling infrequent critical events occurring in dynamic environments.

- Train users to recognize inappropriate uses of the aid and to recognize errors. Provide readily accessible lists of limitations; include information concerning limitations and errors in embedded training. Users should learn not to categorically accept a decision aid's capabilities.

11.4.5 Decision Graphics and Displays

- Prepare graphics, textual reports, and input screens in formats familiar to the user. This will facilitate rapid and accurate information-processing. However, the user should be able to control formats or to select from alternate preprogrammed formats.

- Graphics are another important part of the user interface. Graphics help assist the user in visualizing information. However, guard against inaccurate graphics, as they can have a strong negative impact.

- Provide historical displays of comparative cases, to include time-sequenced presentations.

- Use spatial rather than textual formats when the task involves extensive spatial processing, in particular when task performance time is limited. Use tables rather than graphs when reading specific data points.

11.5 ORGANIZATIONAL FACTORS

11.5.1 Information Requirements

Ensure that decision aids are flexible in meeting the different information requirements at different organizational levels.

- Different levels of organizations require different levels of abstraction. Ensure that decision aids accommodate different levels of detail and time constraints at each echelon.

- Ensure that command and control decision aids are distributed (i.e., they should support multiple, cooperating decision-makers at different locations sharing a common database).

- Where practical, design decision aids to support the entire command and control process, rather than to support isolated phases of the process.

11.5.2 Entire Organization

Ensure that decision aid designs consider impacts on the entire organization, particularly where organizational goals may supersede those of subordinate decision aid users.

11.5.3 Complementary

Ensure that decision aids complement existing tasks and information-distribution systems.

11.6 FLEXIBILITY

11.6.1 Change-over Time

Design decision aids as adaptive systems (i.e., they must accommodate growth and evolve over time to meet changing conditions, doctrine, etc.).

- Establish policies for implementing changes, as well as the mechanisms for those changes.

- Adjust to changing situations and user preferences (different circumstances and users may require different methods).

11.6.2 Maintainability

Ensure that decision aids are maintainable by the user. Rules, data, and decision logic should reflect current needs.

11.6.3 Type of Support

Allow the user to tailor the type of support provided by the decision aid in the presence of changing conditions.

REFERENCES

Paragraph	Reference
11.1.1a	Zachary (1988) pp. 997-1030; McKeown et al. (1991); Ehrhart (1990); Andriole and Adelman (1990)
11.1.1b	Andriole and Adelman (1990); Zachary (1988) pp. 997-1030; Walrath (1989)
11.1.1c	Zachary (1988) pp. 997-1030; Walrath (1989); Andriole and Adelman (1990); McKeown et al. (1991); Ehrhart (1990)
11.1.1d	McKeown et al. (1991)
11.1.2	Zachary (1988) pp. 997-1030; Lysaght et al. (1988); McKeown et al. (1991)
11.1.3.1	Holtzman (1989); Lysaght et al. (1988); McKeown et al. (1991); Walrath (1989)
11.1.3.2	Walrath (1989); McKeown et al. (1991); Holtzman (1989)
11.1.3.3	McKeown et al. (1991); Walrath (1989); Lysaght et al. (1988); Klien and MacGregor (1988)
11.1.3.4	Holtzman (1989); Klien and MacGregor (1988); McKeown et al. (1991); Walrath (1989)
11.1.4.1	Holtzman (1989)
11.1.4.2	Holtzman (1989)
11.1.4.3	McKeown et al. (1991)
11.1.4.4	McKeown et al. (1991)
11.1.5.1	Walrath (1989)
11.1.5.2	Walrath (1989)
11.1.5.3	Walrath (1989)
11.1.5.4	Walrath (1989); Gordon (1988) pp. 55-59
11.2.1a	Gordon (1988) pp. 55-59; Thierauf (1988); Main and Paulson (1988)
11.2.1b	Thierauf (1988)
11.2.1c	Main and Paulson (1988)

REFERENCES (cont'd)

Paragraph	Reference
11.2.1d	Gordon (1988) pp. 55-59
11.2.2	Thierauf (1988)
11.2.2a	Thierauf (1988); Holtzman (1989); Schwartz (1983) pp. 13-17; Andriole and Adelman (1990); LeMay (1988) pp. 227-229; Finke and Lloyd (1988) pp. 170-193
11.2.2b	Urban (1990); Andriole and Adelman (1990); Thierauf (1988); McCann (1988); Main and Paulson (1988)
11.2.3	Urban (1990); Andriole and Adelman (1990); Thierauf (1988); Walrath (1989); Tannenbaum (1990) pp. 54-59; Bidgoli (1989) pp. 27-34; Zachary (1988) pp. 997-1030; O'Keefe (1989) pp. 217-226; LeMay (1988) pp. 227-229
11.2.4a	Minasi (1990) pp. 13-15; O'Keefe (1989) pp. 217-226
11.2.4b	Holtzman (1989); Schmitz et al. (1990) pp. 29-38; Pew (1988) pp. 931-940; Holtzman (1989)
11.2.4c	Holtzman (1989); Bidgoli (1989) pp. 27-34; Thierauf (1988); Pew (1988) pp. 931-940; Finke and Lloyd (1988) pp. 170-193
11.2.4d	Andriole and Adelman (1990)
11.2.4e	Finke and Lloyd (1988) pp. 170-193
11.2.4f	Finke and Lloyd (1988) pp. 170-193; Minasi (1990) pp. 13-15; Tannenbaum (1990) pp. 54-59
11.3.1a	Lysaght et al. (1988); McKeown et al. (1991); Schmitz et al. (1990) pp. 29-38; Pew (1988) pp. 931-940; Ma et al. (1989) pp. 996-1012
11.3.1b	Holtzman (1989); McKeown et al. (1991); Gordon (1988) pp. 55-59; Riedel (1988); McCann (1988); Zachary (1988) pp. 997-1030; Thierauf (1988); Walrath (1989); Ehrhart (1990); Pew (1988) pp. 931-940
11.3.1c	McKeown et al. (1991); Osborn and Zickefoose (1990) pp. 28-35; Lysaght et al. (1988)
11.3.1d	Holtzman (1989); Riedel (1988)
11.3.2a	Gordon (1988) pp. 55-59; Riedel (1988); McCann (1988); Bidgoli (1989) pp. 27-34

REFERENCES (cont'd)

Paragraph	Reference
11.3.2b	Gordon (1988) pp. 55-59; McCann (1988); Urban (1990); Zachary (1988) pp. 997-1030; Minasi (1990) pp. 13-15
11.3.3a	Lysaght et al. (1988)
11.3.3b	McCann (1988); Ehrhart (1990); Holtzman (1989)
11.3.4a	Pew (1988) pp. 931-940; Ehrhart (1990); Schwartz (1983) pp. 13-17; Riedel (1988)
11.3.4b	Holtzman (1989); Gordon (1988) pp. 55-59; Pew (1988) pp. 931-940
11.3.4c	Holtzman (1989); Ehrhart (1990); Pew (1988) pp. 931-940; Tannenbaum (1990) pp. 54-59
11.3.4d	Holtzman (1989); Ehrhart (1990); Finke and Lloyd (1988) pp. 170-193; Riedel (1988)
11.3.5a	Riedel (1988); Crolotte et al. (1980) pp. 1216-1220
11.3.5b	Riedel (1988)
11.4.1	McCann (1988)
11.4.1a	McCann (1988)
11.4.1b	McCann (1988)12.4.1.3; McCann (1988)
11.4.2	Weingaertner and Levis (1988) pp.195-201
11.4.3	Weingaertner and Levis (1988) pp. 195-201; Riedel (1988)
11.5.1	Thierauf (1988); Schmitz et al. (1990) pp. 29-38; Gordon (1988) pp. 55-59; McCann (1988)
11.5.1a	McCann (1988)
11.5.1b	Holtzman (1989)
11.5.2	Bidgoli (1989) pp. 27-34
11.5.3	Urban (1990); Gordon (1988) pp. 55-59
11.6	Mittal (1985) pp. 32-36
11.6.1	Pew (1988) pp. 931-940

REFERENCES (cont'd)

Paragraph	Reference
11.6.1a	Osborn and Zickefoose (1990) pp. 28-35; Mittal (1985) pp. 32-36; Schwartz (1983) pp. 13-17; Riedel (1988); McCann (1988); Bidgoli (1989) pp. 27-34; Zachary (1988) pp. 997-1030
11.6.1b	Pew (1988) pp. 931-940
11.6.2	Zachary (1988) pp. 997-1030
11.6.3	Barnett (1990) pp. 1552-1556

12.0 QUERY

A Database Management System (DBMS) is composed of computer software that facilitates processing information into organized or summarized groups. A database consists of interrelated data that are searchable by a computer. Retrieving information from a database using the DBMS involves identifying a set of items that match or are similar to the user's query or statement of information need. The term "data access" refers to the process of locating and retrieving requested sets of data. By contrast, the term "data presentation" refers to the process of displaying that data to the user in an appropriate fashion. Data access is the query, and data presentation is the result.

The software that makes up the DBMS user interface usually consists of applications programs, report program generators, and query languages. The applications programs allow the end users to enter, retrieve, and update the data in the database. Report-generator utility programs help users specify the content and format of reports. Query languages are used to meet requests for information or to provide a means to browse through the database.

Databases are usually searched in a series of steps. The computer- readable message containing the search terms and logical operators for combining them must be derived from the search query or queries submitted by the user. The search terms are then matched against terms in the database file, either indirectly by searching the index or by directly searching records. The computer responds with counts of retrieved items and should allow the user to sample the items by displaying them on the screen. The user can then make iterative adjustments, either to broaden or narrow the scope of the query.

This section initially reviews types of database queries and methods used to store data in databases. Then, the section provides general guidance on user-oriented database design. The remainder provides specific guidance on query screen designs, user requirements, user-friendliness, database searching, and design requirements for novice and expert user interfaces.

TYPES OF QUERIES

Users most often communicate with databases by means of command-driven (i.e., query languages), form and menu-driven, natural-language, and icon-based interfaces.

Command languages provide flexibility and relieve the experienced user of the requirement to traverse an entire menu structure to select a command. Users of this type of interface must be familiar with the command language, the steps required for solving problems, and the computer's syntax for accomplishing each step of the process. Structured Query Language (SQL) and Query by Example (QBE) are commonly used languages that perform similar functions.

SQL is a textual language that is becoming a relational database standard. SQL includes table definition, database update, view definition, and privilege-granting, in addition to query facilities. SQL is often embedded in programs written in other languages, where it generates query results that can be processed by programs written in the host language.

QBE is a table-oriented version of the SQL relational database language and is often supported where SQL is used. QBE provides a pictorial representation of database tables. Symbols placed in the proper table columns specify query selection conditions, grouping, data display, and database updates. Although QBE's tabular format offers advantages to users, it requires user sophistication for effective use.

Query By Forms (QBF) presents the user with data-entry forms that also can be used as templates when developing queries. When accessing data, the user can select one or more of the data-entry fields and enter values, ranges of values, or logical conditions, which are then automatically translated to database queries. Using familiar forms for data entry and search tasks facilitates the user's performance in creating straightforward queries.

Menus provide user-friendly interfaces to command languages, such as SQL. They are designed in a hierarchical or tree structure, which allows the user to proceed step by step through the menu structure to the desired level of detail. Some menu systems allow the user to go directly to a specified level by keyboard command or selecting items from a multi-level menu map. Menu-based query aids offer several advantages. They lead the user through the problem-solving process by indicating which options are available at each point. They are relatively easy for a novice to use, particularly when unfamiliar with the query command structure (low syntactic knowledge) or uncertain how to proceed in solving a particular problem (low semantic knowledge). Menu systems also have disadvantages. Users are forced to make selections from the choices offered by the system and are, therefore, subject to any constraints that might be present. If a user makes an incorrect choice at any level, it can be time-consuming and frustrating to retrace the steps in the menu structure.

Natural language interfaces allow users to formulate queries in their native language (e.g., English, Spanish). These interfaces use a knowledge of syntax (grammar) and semantics (meaning) to interpret queries and translate them into the query language used by the database system. This approach frees the user from learning the usual conventions and rules of query language. Although natural language interfaces offer great potential, they may require considerable user effort in setting up the underlying dictionaries.

Users may directly query icons, maps, schematics and other visual depictions of physical objects by using a pointing device to select the picture or its features in some sequence. Pointing devices (e.g., a mouse, touch-sensitive screen, or trackball) are often used in combination with menus and text-entry screens to formulate queries. Direct interaction with visual representations of physical objects and icons can facilitate human performance.

DATABASE DESIGN

Ease of use and overall performance of a database system depend on its file structure (the manner in which the records are organized in the file or database) and search processes. The details are chosen by the designer or programmer of the system, often with more concern for the programming aspects of a particular model than for the human performance constraints imposed by that model. The optimum form of information representation will be a function of the task being performed. Unfortunately, current research offers little guidance on how to proceed in database retrieval situations.

Database designs typically use hierarchical, network, relational, or object-oriented models. The hierarchical model represents data in tree structures, and networks represent data as interconnected structures of records linked in one-to-one or one-to-many relationships. Relational databases organize data in tables. Because of its power and ease of use, the relational representation is the prevalent model today and is likely to be the database model of choice in the near future. Object-oriented (sometimes called extended relational) database systems are considered to be part of the next generation of database systems. An object-oriented system represents real-world entities as "objects" that have attributes and defined relationships with other objects.

It is important to recognize that each of these database models can influence the format in which information is presented and the way in which the user can add to, retrieve, or change the information contained in the database. In the end, database models determine the modes of user-database interaction, the format in which the data are presented to the user, and the ease with which a user can acquire information from the database.

12.1 GENERAL RECOMMENDATIONS

12.1.1 Ease of Use

Ensure that a query language or procedure is easy to learn and use. Ease of use and user-friendliness often determine whether the database is used. A program will not be used if it is intimidating, is too difficult, or requires too much effort.

12.1.2 Interactive Queries

Give preference to on-line query over batch or off-line modes because it provides the user the opportunity to interact with the system.

12.1.3 User Assistance

Ensure that an application assists the user in creating complex queries and in narrowing down the search in a step-by-step fashion.

12.1.4 Error Detection

Alert the user to syntax errors in queries and, if possible, to semantic faults (semantic integrity).

12.1.5 Minimum Training

Require only the minimum training. An effective user interface should not require extensive training to be used easily.

12.1.6 User-Oriented Designs

Design the system interface in cooperation with the end users to ensure their satisfaction with the final product. User involvement is most effective when users participate in both developing and implementing the system interface.

12.1.7 Multiple Search Options

Consider the nature of the searches to be performed before choosing an interface format. When more than one type of query is possible, one solution is to choose the interface format that provides the best average performance. Alternatively, provide multiple query and display formats, so the user can change formats as desired or when the nature of the search task changes.

12.1.8 Appropriate Displays

Ensure that displays are appropriate. The forms of information display that facilitate quick responses are not necessarily the same forms that produce accurate responses. The three basic forms of information display are spatial, verbal, and tabular formats. Pictures (spatial) are superior to words (verbal) in recall and recognition tasks and often lead to quicker completion times on procedural tasks. However, words lead to greater accuracy in performance.

12.1.9 Individual Preferences

Ensure, if feasible, that the DBMS is consistent with user expectations. Individual preferences play an important role in the effectiveness of any query application. Users perform better and provide a higher proportion of correct answers when the format of the database matches the format they prefer. Observation shows that, although experience with an application can lead to changes in preference, only preexisting preferences for display formats influenced user performance.

12.1.10 Displaying Results

Display data numerically or graphically. Graphical displays include the bar graph, plot, pie chart and other computer-drawn pictures. Because graphical presentations provide less accuracy than numerical presentations, the most important consideration is the transfer of meaning to the user (see Figure 12-1).

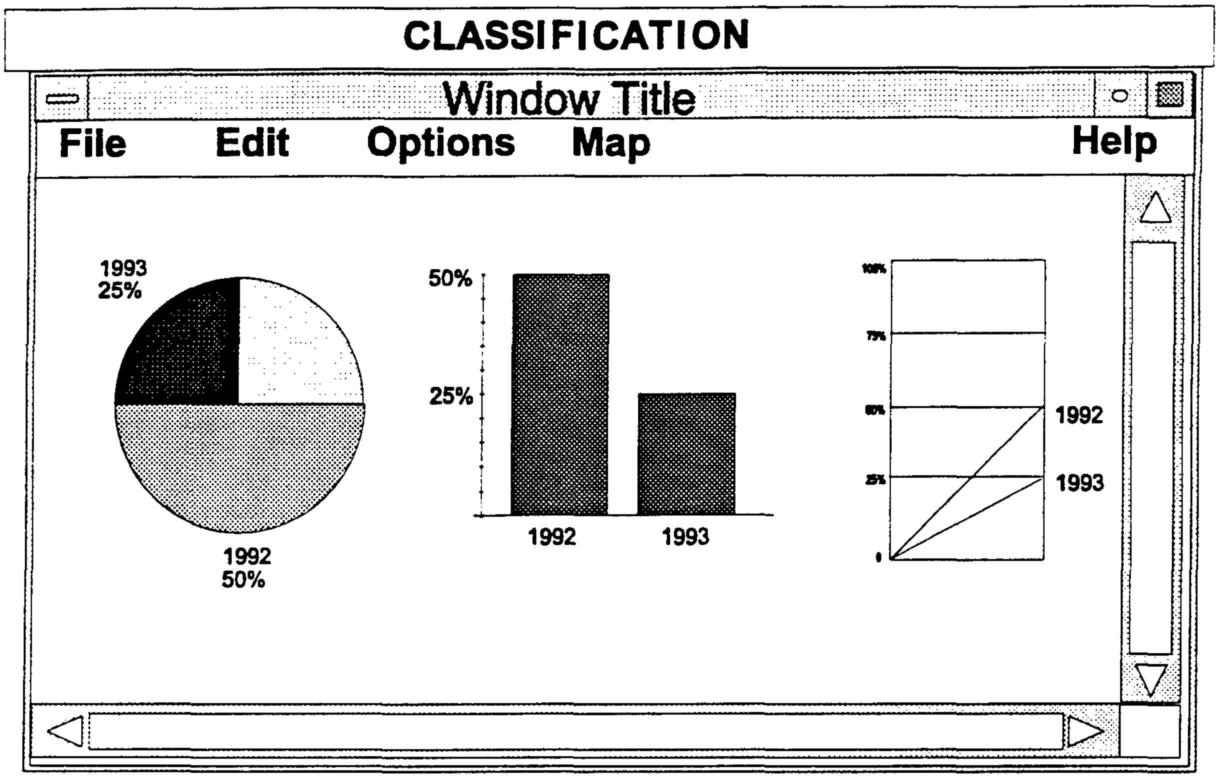


Figure 12-1. Examples of Pie Chart, Bar Graph, and Line Graph

12.2 QUERY SCREEN DESIGN

Query screens display the results of a query request or the contents of computer files. The objective in query screen design is to aid the user in quickly and easily locating data or information. Query screen development should optimize human scanning, as scanning is easier when eye movements are minimized, required eye movement direction is obvious, and a consistent pattern is followed.

12.2.1 Screen Design Principles

- Include on a query screen only information that is relevant to that screen. Forcing a user to wade through volumes of data is time-consuming, costly, and error-prone. If information will never or very seldom be used, do not display it. An item may be relevant one time a screen is displayed but irrelevant another. Limit a transaction or screen to whatever is necessary to perform actions, make decisions, or answer questions.

- Ensure that the interface display groups information in a logical or orderly manner. Locate the most frequently requested information in the upper left corner.

- Locate the most frequently requested information on the initial screens for multiscreen transactions.

- Ensure that the screen is not overloaded, and use spaces and lines to balance the screen perceptually.

- Use consistent terminology, commands, formats, and general appearance throughout the interface. Ensure that learning can be transferred between modules of the program.

12.2.2 Query Screen Organization

Organize the query screen in a logical, orderly, and meaningful manner. When information is structured consistently with a person's organizational view of a topic, that person comprehends more information. Finding information on a query screen can be accelerated by many factors, including the following:

- The interface should locate the most frequently sought information on a screen in the upper left-hand corner. If there are multiple screen transactions, locate the most frequently sought information on the earliest transaction screens.

- To aid the user in locating a particular item, provide easily scanned and identifiable data fields. Accomplish this through columnization with a top-to-bottom, left-to-right orientation, which permits the eye to move easily left to right across the top of the columns to the proper column before beginning the vertical scan.

- Top-to-bottom scanning will minimize eye movements through the screen and enable human perceptual powers to be used to the fullest.

- Current technology presents query output mainly in tabular format. Emerging object-oriented technology will provide different ways to present such information visually.

12.2.3 Captions (Labels)

- Captions should be complete and written in clearly understandable language. Display captions in upper case, although lower case may be used for long, descriptive captions. Do not use reverse video or highlighting for labels.

- For single fields, locate the caption to the left of the entry fields. Separate the caption from the entry field using a unique symbol and one blank space (a colon ":" is recommended). With multiple occurrence fields, locate the caption one line above and centered over the column of data fields.

12.2.4 Data Fields

- The application interface should ensure that data fields are visually distinct from other displayed information (e.g., field labels).

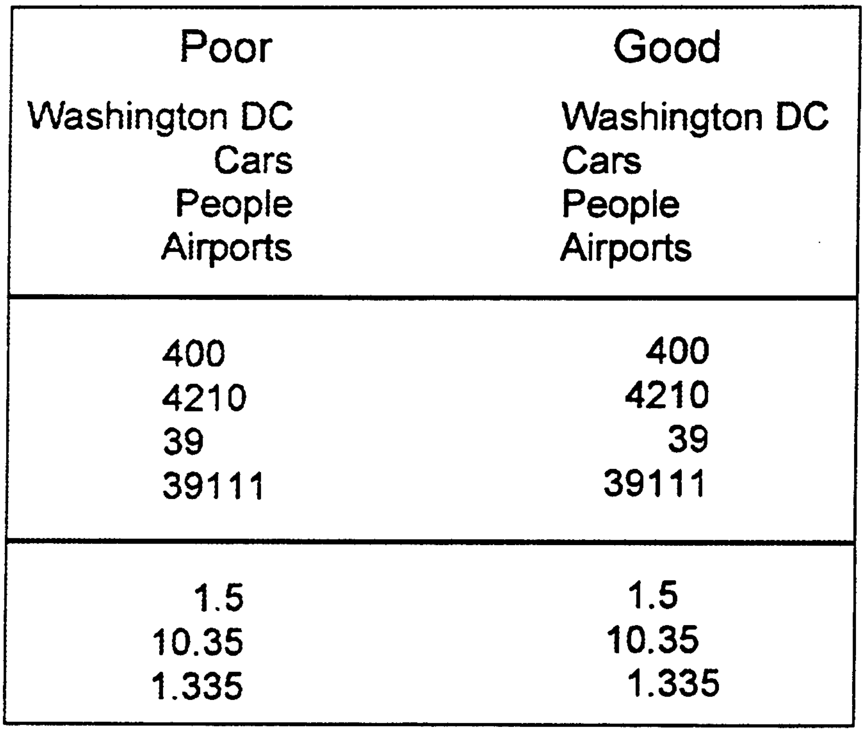
- The interface should display directly usable information, as well as fully spell out codes and compressed information. The data displayed should include natural splits or predefined breaks.

- The interface should display data strings of five or more characters (numbers or alphanumeric) with no natural breaks, in groups of three or four characters with a blank or other delimiter between each group. Data strings should be left-justified, and numeric data should be right-justified or justified about the decimal point. For all types of data, identical data should be consistent despite their origin (see Figure 12-2).

12.2.5 Data Organization

- Organize data in accepted and recognizable order, with vertically aligned captions and data fields in columns.

- Ensure the application justifies data displays consistently.



Poor	Good
Washington DC Cars People Airports	Washington DC Cars People Airports
400 4210 39 39111	400 4210 39 39111
1.5 10.35 1.335	1.5 10.35 1.335

Figure 12-2. Data Layout and Justification

- Promote readability by designing the interface with at least one space between the longest caption and the data field column, and with at least one space between each heading. Section headings should be on-line above related screen fields, with captions indented a minimum of five spaces from the beginning of the heading and fully spelled out.

- When presenting multiscreen transactions, place a screen identifier or page number in the upper right-hand corner of the display (i.e., "screen 2 of 5").
- Locate error and status messages consistently in a separate area of the screen. Emphasize these messages by using a contrasting display feature (e.g., reverse video, highlighting, or preceding series of unique symbols, such as asterisks).
- Provide different forms of information display for different search tasks. For example, the interface should provide a selection of display formats as well as a review format where certain fields of the retrieval records can be reviewed without retrieving the entire record.

12.3 USER REQUIREMENTS

12.3.1 Search Enhancements

- Query optimizers are software procedures that automatically enhance the ability of the database application to execute queries. For example, the computer would initiate the search when the first several characters of the search string were entered to reduce the overall perceived time delay of the search. Use query optimizers to increase the effectiveness of the program, but they should be invisible to the user.
- Allow the user to rank search terms by importance. Then use this ranking in a formula for automatically ranking records by relevance in the retrieval set.
- Provide additional search terms in a retrieval set. For example, use a memo field to list the additional search terms related to a particular field.
- Ensure that the application allows redisplay of results of the previous search without requiring reprocessing.

12.3.2 Automatic Functions

- Provide automatic recognition of spelling variants (e.g., color versus colour).
- Provide automatic recognition of acronyms.
- Provide automatic recognition of variations in romanization (e.g., Peking versus Beijing).
- Provide automatic inclusion of the inverted form (e.g., Newborn Infant to Infant Newborn).
- Ensure that the application automatically removes punctuation from search terms when matching them against search-key values.

12.3.3 Word Stemming

- Ensure the application uses a set of rules for reducing words to their root forms by stripping them of their suffixes (e.g., reduce, reduction, reducing).

- If desired by the user, ensure that the application automatically searches the index for all words containing a given root (e.g., the word "form" is the root of formation, inform, and information).

- Provide rules for exceptions based on the language of the discipline or specialty area.

- Allow truncation. The application should automatically search for all words or phrases that begin with the same character stem (e.g., term for terms, termination, and terminated).

12.3.4 Erasing

- Allow immediate deletion of individual characters or deletion of the entire line of input (provided it has not been processed by the computer).

- Permit deliberate interruption of computer messages or displays without disconnection (break or interrupt key).

12.3.5 User Satisfaction

- User satisfaction with the system can be enhanced by including the factors described in the following paragraphs.

- Provide results in a timely manner. One factor of timeliness is the elapsed time from when the command is sent until a response is displayed (response time). Another is the time required for characters or graphics to appear on the screen or hard-copy device (display rate).

- Ensure the appearance, print format, and organization of output are natural to the user. User-generated report formats aid in matching the appearance of the output to what the user expects.

- Minimize the level of effort required by the user, including the limitations or qualifications that the application places on search output.

- Provide maximum capability to the search system while maintaining maximum retrieval effectiveness. For example, do not increase the database size to the point where retrievals take excessive time without also improving search methodology to compensate.

- Ensure that the application assists the user in formulating searches for maximum usefulness of the search results.

12.4 USER-FRIENDLINESS

12.4.1 Commands

- Use mnemonics to avoid the need for remembering syntax (i.e., as sequences or specifications in output instructions).

- Use commands in an easy-to-learn, user-oriented system language.

- Use unambiguous commands. The meaning should be clear to the user.

- Ensure that entering data is not physically awkward for the user, and keystrokes are limited to those absolutely necessary. Provide the capability to define Ctrl key, Alt key, or function key combinations (i.e., Ctrl/Alt/Del to reboot the system) in place of keystroke combinations.

- When a command will delete stored information, provide a complementary command that reverses the action. If deletion is irreversible, provide the user with the opportunity to reconsider the action. The application should check for meaningless commands against a list of authorized commands, after which the application should allow the user to enter a revised command rather than automatically abort the procedure.

- Provide the user with abort or escape facilities for controlling the dialog flow.

12.4.2 Computer Messages

Messages should be clear, simple, and concise. Present the user with the briefest message that can be properly interpreted. Directive messages should be specific and in the context of the current working environment. Messages should warn the user of irreversible action.

12.4.3 Error Messages

Deal with mistakes in a positive, helpful manner. Users will thus gain confidence in the system and feel less intimidated or fearful of damaging it or the data. The error message should appear when the user enters a command that is misspelled, improperly formatted, or cannot be processed because it is inappropriate to the situation. The message should provide instructions for revising the erroneous command.

12.4.4 Documentation

Full system documentation should be available in manual form.

12.4.5 Tailor the Interface

Tailor the interface to suit the needs of users.

- Tailor frequently used queries. In cases where the value of only one or two parameters changes, provide the user with default values for those parameters. For example, a query might request the names of all *Army officers* with *over ten years service* who *graduated from an academy* in *the top 10 percent* of their class and who *serve in the infantry*. This query contains elements that could be requested several different times, using slightly different conditions each time.

- Macro definition procedures (user-defined commands) are an important feature for expert users who prefer to define their own commands and personalize their environments by encapsulating frequently used query sequences in a new command. Macros greatly simplify user interactions with the application as well as save time. The application should allow the user to store these macros as files or define function key combinations to perform the function.

12.4.6 Accelerators

Ensure that the interface provides accelerators to save keystrokes. For example, special keys can be dedicated to commonly used functions. The application should permit direct commands as alternatives to menu options.

12.4.7 Backup

Ensure that the application shields the user from system failure. Provide backup facilities both internally by the software application program and externally by the operating system.

12.4.8 Restore

Ensure that the application provides a restore utility to facilitate recovery of damaged or destroyed data from backup copies.

12.4.9 Interrupt

Ensure that the system provides the capability to interrupt work with the application software, then comes back later to resume work at the same point.

12.5 SEARCHING

12.5.1 Commands

Make the following types of database utility and search commands available to the user.

- Provide a database SELECT command.

- Provide commands to create and erase sets.

- Allow users to combine two or more sets to create new sets.

- Provide the capability for users to specify report formats. The user should be able to name the report, identify the relations from which the report data will be derived, determine the report layout, and define the lines and headings or captions of the report. The user should be able to save the created formulating query and report format for later use.

- Provide the capability for users to restrict the output of retrieval sets.

- Provide the capability for users to save search results easily.

- Provide the user a list of previous search commands upon request. The number of saved commands could be set by the user or could be a prespecified number.

12.5.2 Control Functions

Ensure that the application provides control functions to aid the user in dealing with the system. These functions should include signaling about the system's current state or performing an action based on the state.

- The input parameter for the MARK command should be the current field value, and the application should note the marked value for future reference. For example, fields or records could be marked for deletion.

- DESCRIBE should use as its parameter the current field value. Provide the user with a detailed explanation or description.

- The parameter for the DROP command should be the current field value. The current field should be dropped from the structure.

- The application should provide the user status information upon request. This should include the completion and success or failure of the last search operation executed.

12.5.3 Editing Commands

Editing commands are necessary during query formulation. The application should provide a text-editing box to be used for typing search queries. The following functions should be available (see Figure 12-3).

- CUT should allow the user to remove the selected text and place it in a clipboard.

- COPY should allow the user to duplicate selected portions of text and place them in a clipboard.

- PASTE should allow the user to place text from the clipboard into the current text.

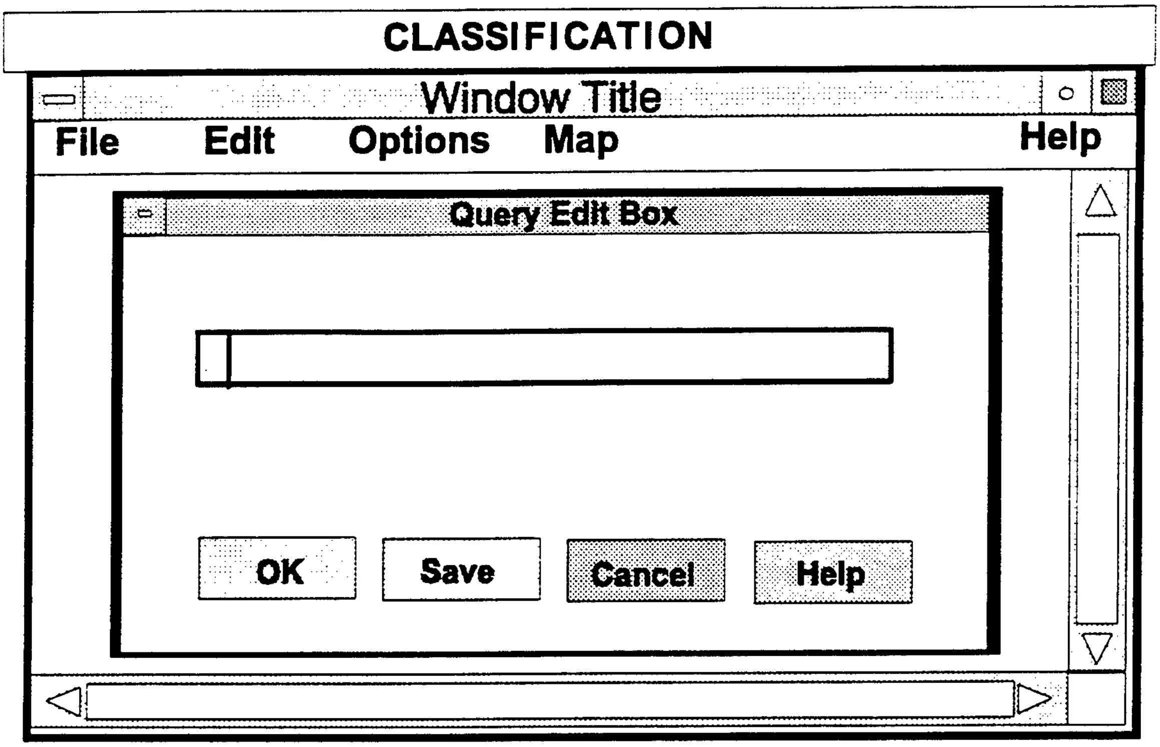


Figure 12-3. Sample Text Editing Box

- CLEAR should remove all characters currently in the text-editing box.

- SEARCH should allow the user to locate a word or group of characters in the text-editing box.

- When used in conjunction with SEARCH, REPLACE allows the user to replace a word or set of characters with another word or set of characters.

- SPELL CHECK should check the words in the text-editing box against a dictionary of recognized words. This function also should check textual commands to assure correct spelling and syntax.

12.5.4 Query Formulation Commands

Major tasks performed by queries include extracting, manipulating, and performing calculations on tabular data, including creating tabular results and new tables. Query applications should be able to build functions as needed for developing application programs, as well as update and maintain tables.

- SELECT should provide a means of identifying fields to appear in the query results.

- COMPILE should generate an executable function and check for correctness.

- RUN or DO QUERY commands cause execution of the query. The application should monitor the execution with prompts for input and error recovery.

- The SHOW command should allow various presentations of a tabular result and could be used to present a preview of the results of a query or report.

- MODIFY should allow the user to make changes to the query definition of an already existing query or report. The new query could be saved to a file or as a report, if desired.

- SAVE should allow repeated use or modification of a query. Store the queries in a file with a unique extension, such as ".QRY."

12.5.5 User-Friendly Searching

- Abbreviations should be significantly shorter than the original word or mnemonic. Truncation is the preferred form of abbreviation. The application should allow both the abbreviation and the full term.

- The application should automatically complete a search term (opposite of truncation) as soon as it recognizes that the portion of the term entered is unique in the index of search terms. The application should stop the user from typing once the search is uniquely identified.

- Because even simple queries can overload the computer, the computer should inform the user of the problem and prompt user input to terminate the query or continue.

12.5.6 Features

- Provide an interactive program that allows the user to navigate through the database. The BROWSE function is especially helpful when queries would be too lengthy to run interactively.

- Provide facilities to format the results of queries as reports.

- Ensure that the application provides the ability to view the list of words and phrases available for searching and term variations, including a link to a database thesaurus to suggest search terms.

- Parsing is the process of deciding how the field will be entered into the search index. Parsing decisions have a direct impact on how a database can be searched. Provide flexibility in searching, regardless of how fields were parsed.

- Use proximity searching, which provides the ability to search words in a positional relationship from word index fields such as titles or abstracts. The words should be either in a specific order or independent of order. For example, the words "query" and "formation" could be searched in the same field.

- Ensure that the application provides the use of Boolean logic, including the use of the logical operators AND, OR, and NOT. It should prompt the user for sets consisting of search terms and combine (intersect) the sets. The search can then be completed as a combined (union) set. The application also should allow interactive editing of queries.

- Ensure that the application provides set building as a means of performing the search in a series of steps, then views the records that answer a query as a set defined by the query.

- Include range searching in the application. This type of search should be based on an ordered sequence using FROM and TO.

- Ensure that the application allows the user to specify the fields to search, because limiting a search to particular fields may speed the search.

- Use a controlled vocabulary of natural language terms. This helps novice users formulate queries.

- Ensure that the application facilitates selecting search terms from key words in records. Then, the interface can display these terms and prompt user selection. For example, the application could rank additional search terms by frequency of appearance in a retrieval set and provide them in ranked order.

- Provide the capability for the system to search on specific data field values input by the user. The application should provide a list of possible field values from which users select.

- Ensure that the application is able to order the field values in a reasonable way, such as alphabetically or from greatest number to the smallest.

- Ensure that the application provides a crossfile search, which will obtain the number of references in all potential databases for the search terms or search profile.

12.6 MULTIPLE LEVELS

User-friendly features and requirements differ for the novice and experienced user. Because the novice will become a more experienced user, the HCI needs to change to suit the evolving needs of the user and the demands of users who have different levels of expertise.

12.6.1 Accommodate Novice and Experienced Users

Multiple levels of interaction are necessary to accommodate the varying levels of experience.

- Users should be able to change levels at any time during a session.

- A tutorial mode should be available when possible.

- Offer context-sensitive HELP on request at all levels.

12.6.2 Novice Users

The level of computer knowledge required of a novice user should be minimal. The application should be easy to use, provide familiar terminology, and allow the user to begin work with little training. To be used effectively, an application should not depend on a complex command language. However, this ease of use may require a loss of power and flexibility.

- An interface for novices may contain only a subset of the search capabilities. This system may be a scaled-down version of a more comprehensive program. In addition, these interfaces may require fewer searchable fields, so the system may not attain the same specificity or variety of search techniques.

- The computer software should prompt the novice user to select from a list of options. The interface should provide an explanation of the options presented.

- The interface for novices should have a simplified command structure using fewer and more easily understood commands.

- Mnemonic selections are preferred over numbered selections.

- The system design should strive for intelligent interfaces between naive users and search systems. Two main components of an intelligent front-end are forms and graphics. Menus and data forms can control the flow of the application, and graphics can be used to provide a visual readout of the data.

12.6.3 Experienced User

Experienced users can accommodate comprehensive versions of query applications.

- The application designed for the expert user can reduce computer overhead by providing less detailed on-line information.

- The application should allow the experienced user to enter multiple commands to speed the dialog.

REFERENCES

Paragraph	Reference
12.0	Hansen and Hansen (1992); Flynn (1987) p. 221; Harter (1986) pp. 124-127; Frost (1984) p. 8; Humphrey and Melloni (1986) pp. 41-50; Kelley (1984); Katzeff (1986); Martin (1983) pp. 427-483; Hershman, Kelly, and Miller (1979); Schur (1988); Shneiderman in Vassiliou (1984); Boehm-Davis (1989); Ogden and Brooks (1983)
12.1	Martin (1983) pp. 427-483
12.1.1	Tenopir and Lundeen (1988) pp. 43-45
12.1.2	Frost (1984) pp. 187; Martin (1983) pp. 143, 452
12.1.3	Martin (1983) pp. 452
12.1.4	Grill (1990) pp. 78
12.1.5	Cuff (1980)
12.1.7	Boehm-Davis (1989)
12.1.8	Boehm-Davis (1989)
12.1.9	Boehm-Davis (1989)
12.1.10	Flynn (1987) pp. 401-441
12.2	Galitz (1989) pp. 155-165; Flynn (1987) pp. 401-441
12.2.1e	Tenopir and Lundeen (1988) p. 45
12.2.5f	Boehm-Davis (1989)
12.3	Humphrey and Melloni (1986) pp. 200-203
12.3.1a	Chapnick (1989)
12.3.3d	Humphrey and Melloni (1986) pp. 143-144, 200; Tenopir and Lundeen (1988) pp. 34; Ehrenreich (1982)
12.3.5	Harter (1986) pp. 154
12.3.5a	Shneiderman in Vassiliou (1984) p. 5
12.4	Tenopir and Lundeen (1988) pp. 44-45; Humphrey and Melloni (1986) pp. 197-199
12.4.1d	Flynn (1987) pp. 520-521
12.4.1e	Flynn (1987) pp. 514-515

REFERENCES (cont'd)

Paragraph	Reference
12.4.1f	Benbasat and Wand (1984); Carlson and Metz (1980)
12.4.4	Tenopir and Lundeen (1988) pp. 155-167
12.4.5	Frost (1984) pp. 241-242; Feldman and Rogers (1982)
12.5	Harter (1986) pp. 76-94
12.5.1	Harter (1986) p. 28
12.5.2	Lochovsky and Tsichritzis in Vassiliou (1984) p. 131
12.5.4	Schauer in Blaser and Zoeppritz (1983) p. 33
12.5.5	Humphrey and Melloni (1986) p. 200; Tenopir and Lundeen (1988) p. 34; Ehrenreich (1982)
12.5.5c	Grill (1990) p. 78
12.5.6	Tenopir and Lundeen (1988) pp. 31-39
12.5.6a	Frost (1984) pp. 196-197
12.5.6c	Sormunen in Wormell (1987)
12.5.6j	Humphrey and Melloni (1986) p. 200; Harter (1986) pp. 41-58; Ogden and Brooks (1983)
12.5.6k	Humphrey and Melloni (1986) p. 202
12.5.6n	Sormunen in Wormell (1987)
12.6 .	Humphrey and Melloni (1986) pp. 199-200; Benbasat and Wand (1984); Tenopir and Lundeen (1988) p. 45; Shneiderman in Vassiliou (1984) pp. 4, 7; Harter (1986); Schur (1988); Kelley (1984); Flynn (1987) pp. 504-506; Cuff (1980); Sormunen in Wormell (1987)

13.0 EMBEDDED TRAINING

The interface of the optimally designed computer program should be designed and tested such that no user assistance is needed. However, because of differences between humans and computers, the variety of task demands, and the ever-present human tendency to make errors, assistance is needed. People differ in computer experience, patience level, learning style, reasoning ability and style, and numerous other characteristics. At the same time, sophisticated computer systems and software programs are often highly complex but still retain the requirement to be highly usable without requiring extensive training or technical expertise. Assistance programs offer one of the primary methods used by designers to achieve a high degree of usability.

User assistance is commonly offered through on-line help, documentation, and on-line training. The distinction between on-line help and on-line training is often blurred. For the purposes of this *Style Guide*, on-line help refers to assistance for a specific problem, function, command, or term. On-line training programs focus on process; they offer instruction.

On-line training programs may exist completely embedded within the application software, separately as an application, or as a combination of both. The on-line training program also may be executed by some form of supplemental component (e.g., strap-on [video disk player] or plug-in [floppy disk]). Though many guidelines apply to both embedded and supplemental training, interface guidelines presented in this section pertain specifically to embedded training.

The guidelines also apply to a range of embedded training formats and capabilities including:

- Fixed format provides the same information regardless of what the user has done.
- Context-sensitive format depends on what users are currently trying to do or on the context in which they are working
- Prompting intervenes or prompts automatically if a user proceeds incorrectly.
- Dialog allows users to obtain assistance through natural language interaction.
- Adaptability keeps track of a user's operation and provides appropriate help or training based on the user's operation, for example, intelligent tutoring systems (Kearsley 1986).

The embedded training guidelines included in this section are derived from the results of empirical research, reported computer training experience, and experts' recommendations. Guidance for embedded training interface design appears under a variety of types of on-line training: Computer-Assisted Instruction (CAI), Intelligent Computer-Assisted Instruction (ICAI), Computer-Based Training (CBT), Intelligent Tutoring Systems (ITS), Embedded Training (ET), coaching, Electronic Performance Support Systems (EPSS), and guided discovery, among others.

On-line training strives to support learning how to use an application. However, it conflicts with the user's primary task, because consulting a training program interrupts work in progress. This conflict may cause new users to skip training altogether or to select immediate task help without furthering their overall understanding of the system (Grice 1989; Hackos 1991; Horton 1990). Two crucial factors in determining whether or not users accept and use embedded training are a well designed, intuitive interface and the opportunity to practice. Each of the embedded training guidelines addressed in this section assumes a basic set of objectives for assisting users: consistency, efficient use of capabilities, minimal memory load on users, minimal learning time, and flexible support of different users.

The goal of embedded training interface design is to ensure users can obtain answers to their questions with maximum efficiency, maximum accuracy, and minimum additional memory requirements. An embedded training program should answer the following types of questions:

- Goal-oriented: What types of things can I do with this program?
- Descriptive: What is this? What does this do?
- Procedural: How do I do this?
- Interpretive: Why did that happen? What does this mean?
- Navigational: Where am I?
- Choice: What can I do now?
- History: What have I done? (Baecker and Small 1990; Gery 1991; Laurel 1990).

The manner in which assistance is provided affects the ability of users to learn and transfer that learning to other situations. Research in instruction and on-line documentation has identified basic concepts and practices that support learning and transfer. Central among these concepts relating directly to embedded training are:

- Opportunity to practice
- Readability
- User control - perceived and actual
- Learning mode - visual (graphics and text)
- Advance organizers.

Specific guidelines related to these concepts appear in embedded training components and instructional presentation guidelines (see Subsections 13.4 and 13.6).

Much of the relevant research and development work from which these guidelines were developed comes from individual demonstration and limited distribution systems. Significantly less empirical research has explored the behavioral issues pertaining to on-line training or advice-giving systems. On-line training experts suggest additional behavioral research, including effects of feedback timing, preferences for and effectiveness of different on-line training components, suitability of media presentation (animation, text, sound), and the effects of system-initiated intervention. The current research focus has moved from building systems capable of detecting all possible errors and misconceptions to building an empathetic partner that chooses among several forms of interaction based on the content of the task and needs of the user.

13.1 GENERAL

A strong embedded training interface provides users with an understanding of the training program and the linkage between the application program and the training program. The guidelines in this section address user orientation and the linkage between application and embedded training programs. Section 13.0 of the *Style Guide* includes general guidance for embedded training, followed by guidelines pertaining to more specific features. This section also contains a series of figures illustrating embedded training. Each figure is based on a basic screen prototype (see Figure 13-1). To illustrate a particular guideline clearly, a number of the figures show only a portion of the basic screen display.

13.1.1 Initial Use Overview

Provide first-time users of embedded training an overview of the embedded training program. This orientation should convey what the embedded training achieves by combining text and graphics (animated or static).

13.1.2 Positive User Attitude

Build positive user attitudes and increase use of the embedded training by ensuring that the interface maintains a positive tone and does not evaluate the user's performance when practicing and experimenting. Ensure the system messages do not blame the user and avoid implying that the computer is human. For example, "You can use the training Program to learn..." is preferable to, "The training Program can teach you..."

- Do not use personalized messages, as they interrupt and often annoy users.
- The effectiveness of the embedded training is related directly to the accuracy of the embedded training information.
- Avoid personalization (i.e., "You did a good job, Sam") and personal recognition (including even simple statements, such as "Excellent!").

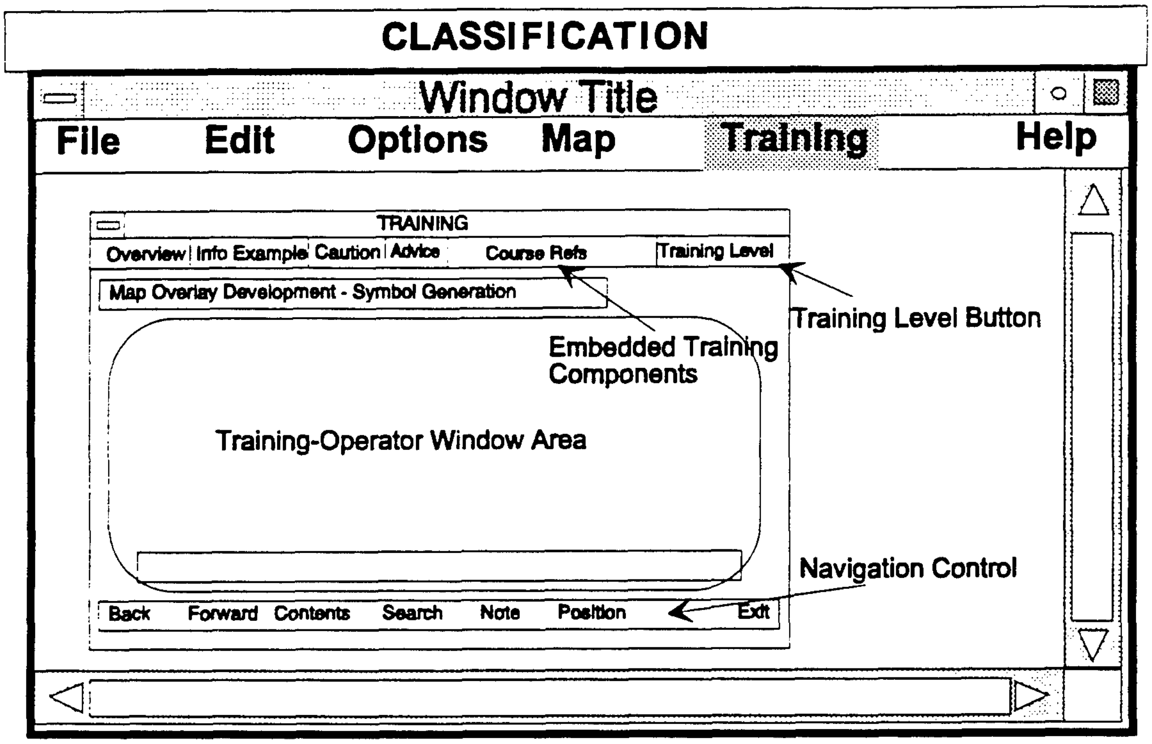


Figure 13-1. Simplified Prototype Embedded Training Screen

13.1.3 Availability of Embedded Training

Provide embedded training programs to users at all points during use of the application, except where it would interfere with time-critical operations.

13.1.4 Accuracy of Embedded Training

Ensure embedded training is accurate, reflects the most current form of the application, and is updated in response to changes in the application. When changes occur in application procedures or critical operations, ensure that users are notified. In addition, consider providing users with an option to see new and revised information (i.e., by selecting "News").

13.1.5 Moment of User Need

Provide training support at the moment the user needs it whenever possible.

13.1.6 Embedded Training Browsing

Allow users to work with the embedded training independent of the application, to accommodate user browsing.

13.1.7 Return to the Application From Embedded Training

Ensure users can return to the application from any point within the embedded training with a single action (e.g., keystroke, command, point and click) without shutting down either system.

13.1.8 Application Restore Screen

When users exit the training program, restore the application screen to the state that existed prior to the request.

13.1.9 Restore Embedded Training

When training is interrupted (e.g., system failure, time-critical requirements) or users exit before completion, offer them the opportunity to return to the position in the embedded training that existed before the interruption.

13.1.10 Protection From Hazardous or Destructive Actions

Prohibit users from accidentally activating hazardous events (e.g., mine field activation) and destructive control actions (e.g., accidental erasure or memory dump) during the embedded training.

13.1.11 Application Screen Protection

Ensure embedded training commands do not alter or destroy application screen data.

13.1.12 Noninterference During Critical Operation

Prohibit system-initiated embedded training interruption of the primary application during a critical operation.

13.1.13 Notification of Critical Operation

Ensure the user is notified of incoming critical application information (e.g., tactical operation input).

13.1.14 Multiple Stations

If the application system has multiple stations, ensure that stations using the embedded training have no effect on the stations performing an operational task.

13.1.15 Context Sensitivity

Make the training context-sensitive; that is, wherever possible, the training should depend on where the user is in the application or on the general nature of the content of the application.

13.1.16 Consistent Application Interface

Provide the greatest possible consistency between the application interface and the embedded training interface to ensure a smooth transition between platforms and to minimize the user's learning requirements (e.g., terminology, displays, commands).

13.1.17 Inconsistent Interface Assistance

Provide assistance if the embedded training interface is substantially different from the application systems operations or when the embedded training interface has complex features that might need to be explained.

13.2 ADAPTATION TO USERS

Users vary in many ways, including computer experience, domain experience (program content - e.g., command and control), learning style, preferred work style, and immediate task demands. Adapting the embedded training interface to the user's characteristics and preferences will encourage use of embedded training and, consequently, should increase user efficiency with the application (see Figure 13-2).

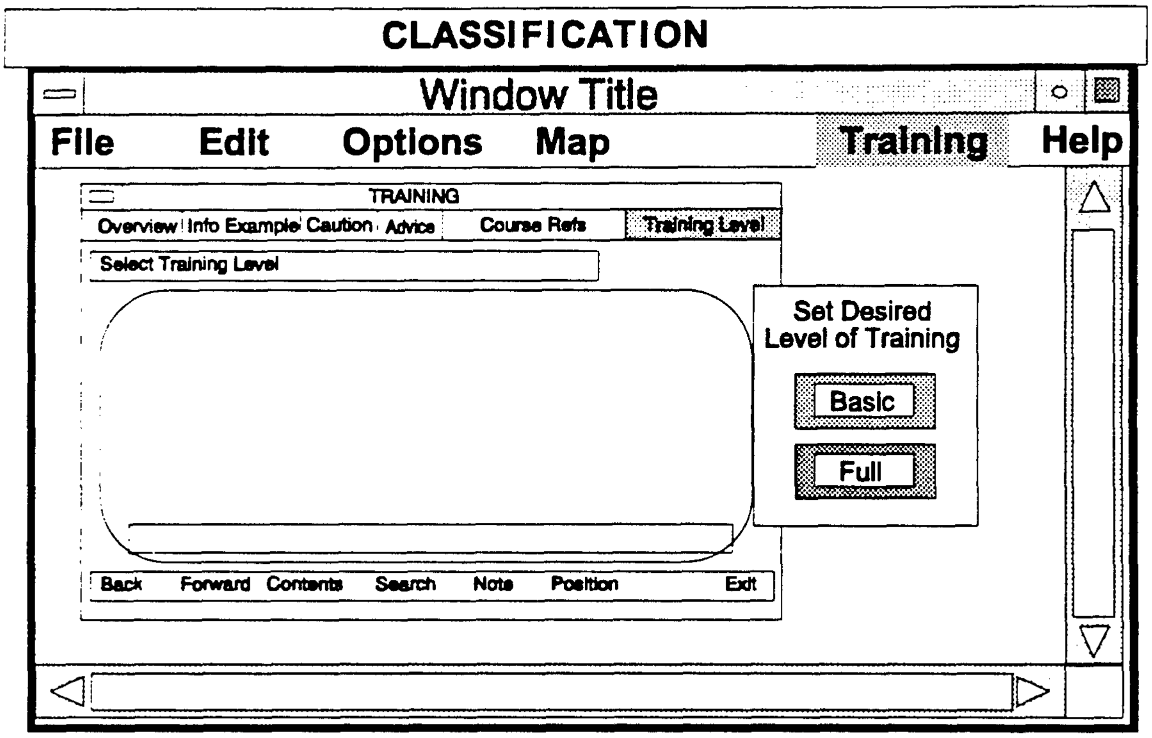


Figure 13-2. User Selection of Training Level

13.2.1 User Control Over Level of Difficulty

Accommodate the differences in computer experience by allowing users to select the level and type of assistance. The ability to select is important because users may be novices in some areas, casual operators in others, and experts in still others. Reducing the complexity of the training interface for beginners simplifies the demands of learning.

- Allow novice users and/or first-time users to select a restricted capability interface that blocks features and allows only basic feature operation (e.g., in word processing - creating, editing, and printing).

- Provide novices only the necessary information, but allow them access to all of the capabilities by direct request.

- Offer experts assistance in using the system more efficiently (e.g., shortcuts, limitations, complex operations).

13.2.2 Learning Structure

Allow the user to select a type of learning structure. This accommodates individual needs for information and practice. Learning structure types may be:

- Discovery - undirected exploration or browsing

- Guided/supported discovery - directed exploration

- Structured - menu identifies options explicitly and provides implicit cues.

13.3 EMBEDDED TRAINING COMPONENTS

An embedded training system can integrate several resource components to support users while they perform their jobs. Embedded training programs may provide an information database (infobase), common errors, examples and scenarios, interactive advice, internal cross-referencing, expert system-initiated training, and formal courseware.

13.3.1 Multiple Components

In addition to the immediate context-sensitive assistance, offer users multi-component training that is easy to specify and access (e.g., scenarios, examples, information databases, off-line references, common problems, and/or coaching).

13.3.2 Information Database Component

Provide users an interactive information database containing both conceptual and task-oriented information.

13.3.3 Reference Component

Provide a reference component that includes all on-line resources, as well as system- and job-related, off-line resources.

13.3.4 Examples Component

Offer users the opportunity to practice using common examples in an exploratory or guided mode, which would allow users to work through the steps required to perform a specific task.

- Encourage user experimentation (i.e., "What would happen if...") by making it easy for them to exit an application, practice, then return to the unaltered application position.
- If users explore a problem within the application, protect the application system with an UNDO command requirement.
- Clearly distinguish between the exercise and the application to minimize possible confusion arising from switching back and forth between operation modes (e.g., highlight or shadow the practice session).
- Avoid demonstrations and exercise summaries if they do not provide opportunities to practice the procedure or function.

13.3.5 Advisor or Coaching Component

Provide an embedded training component that advises or coaches users in solving problems. This may make users aware of enhanced system operation and may also be used in response to user request, system recognition of suboptimal user performance, or complex tasks. A system can coach users through tasks by presenting a series of questions and recommending a course of action based on the responses (see Figures 13-3 and 13-4).

13.3.6 Common Errors Component

Provide users with a context-similar, embedded training component that shows common user errors or "Cautions" associated with a given approach or task procedure (see Figure 13-5).

13.3.7 Record Keeping

Records of user interaction with the embedded training can be helpful to users, supervisors, and system designers. Using embedded training records requires careful planning to avoid threatening users. Users are more likely to experiment and practice if they feel their errors will not be seen by others.

- Allow the user to record the path through a process and/or the training modules completed successfully or unsuccessfully. This will aid in later reference or experimentation.

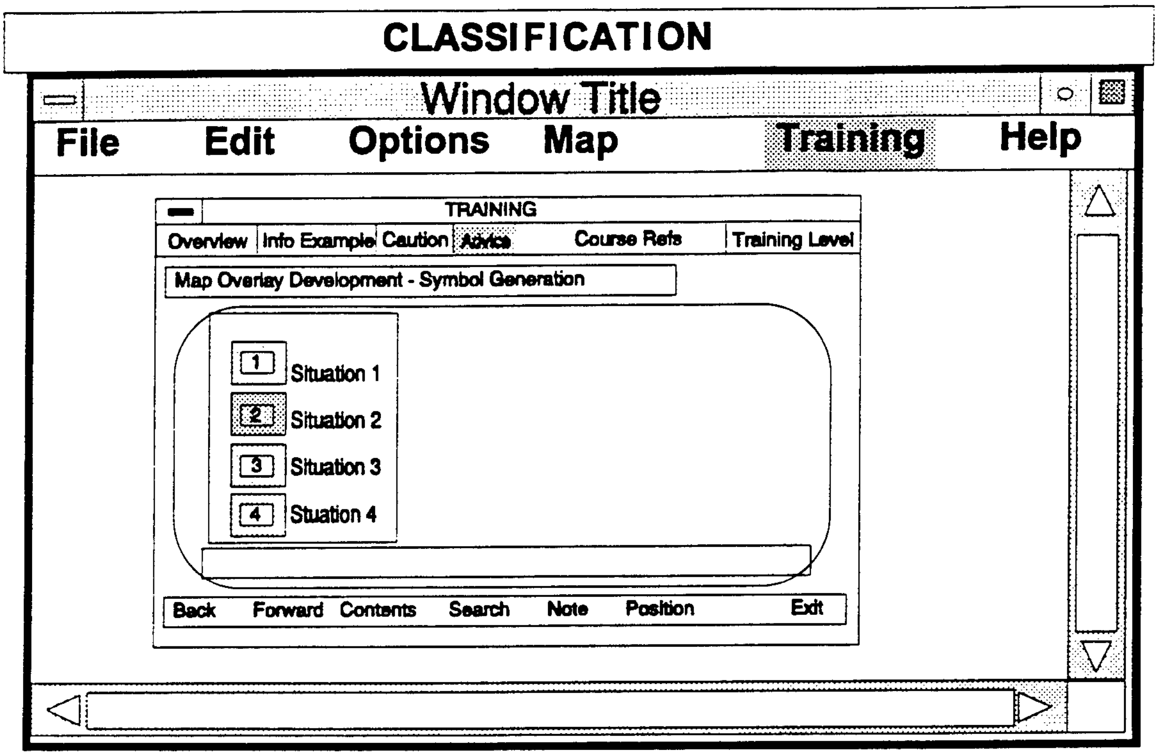


Figure 13-3. Example of How an Advisor Can Guide Users

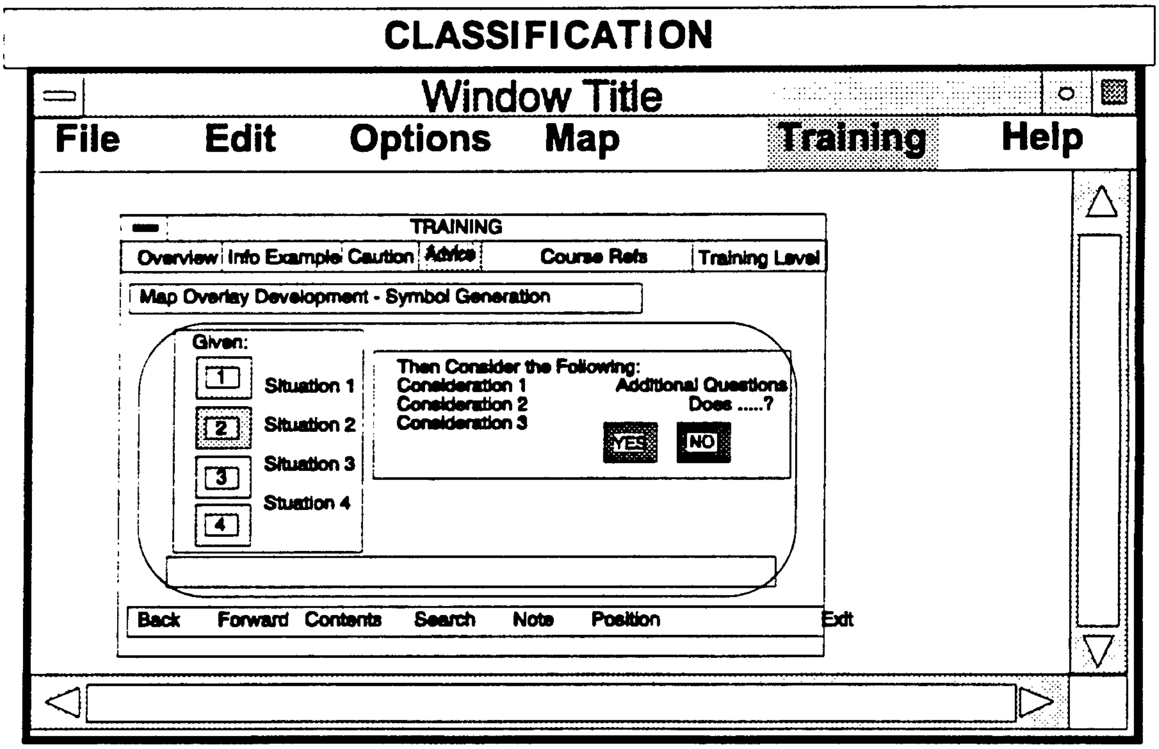


Figure 13-4. Example of How an Advisor Can Guide Users (cont'd)

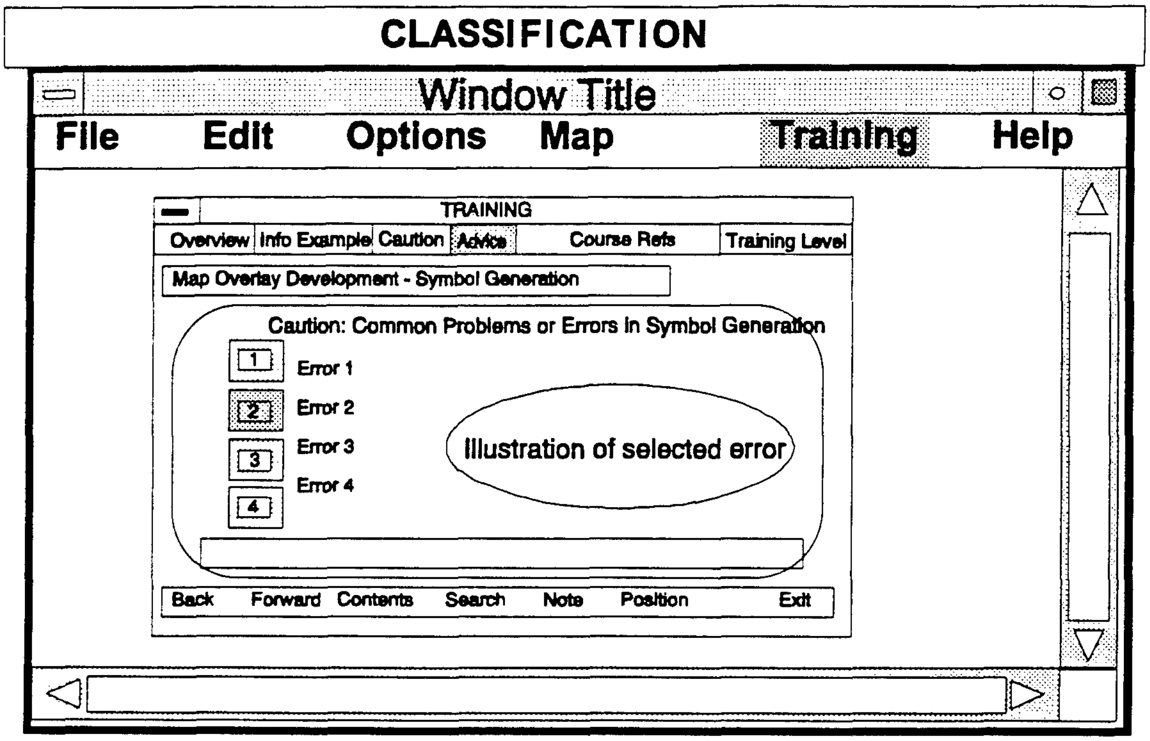


Figure 13-5. Example of "Cautions" that Identify Common Errors

- If records of user training sessions are stored, ensure the privacy of users is protected by storing their records as anonymous files.
- If the exercise or courseware module will be used for evaluation purposes, give users prior notification. Explicitly state the criteria for evaluation.

13.4 INSTRUCTIONAL STRUCTURE

Both the size of instructional unit (granularity) and control of instructional sequence affect the efficiency and attitude of the user.

13.4.1 Granularity

Structure the embedded training components into "single learning episodes," small enough and homogeneous enough to be learned as single units. This enables users to select the particular section or subtopic within a component for which they desire assistance.

13.4.2 System-Controlled Sequences

For novice users and for embedded training that deals with critical or hazardous procedures, the system should direct user movement through the procedures.

13.4.3 Sequence Control for Experienced Users

Provide experienced users with the flexibility to move through the steps of a procedure sequentially or to move directly to any specific step or resource point.

13.5 INSTRUCTIONAL PRESENTATION

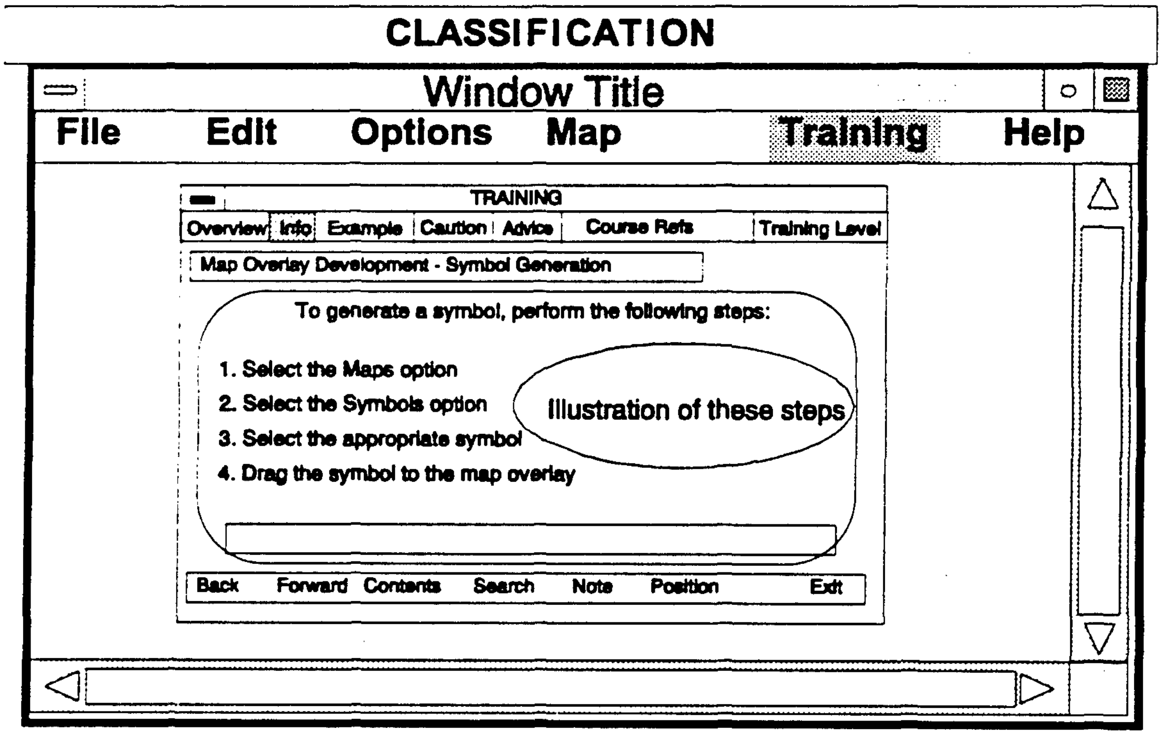
The manner in which assistance is provided affects the ability of users to learn from the instructional experience and to transfer that learning to other situations. The following statements outline interface guidelines for instructional presentation.

13.5.1 Combined Media Presentation

Present the embedded training using a combination of media, graphics, and natural language, where appropriate. Graphic media aid in visualizing significant patterns, whereas natural language text conveys the meaning and significance of the visualization (see Figure 13-6).

13.5.2 Graphics for Method-Based Knowledge

Offer users flowchart diagrams that provide an overview conveying method-based knowledge, consisting of a series of procedural steps and decisions.



Map Overlay Development - Symbol Generation

To generate a symbol, perform the following steps:

1. Select the Maps option
2. Select the Symbols option
3. Select the appropriate symbol
4. Drag the symbol to the map overlay

Figure 13-6. Combined Graphic and Natural Language Presentation

13.5.3 Reading Requirements

Keep reading requirements to a minimum. Users prefer to read text in print and may not read text on a screen that exceeds even a few sentences in length.

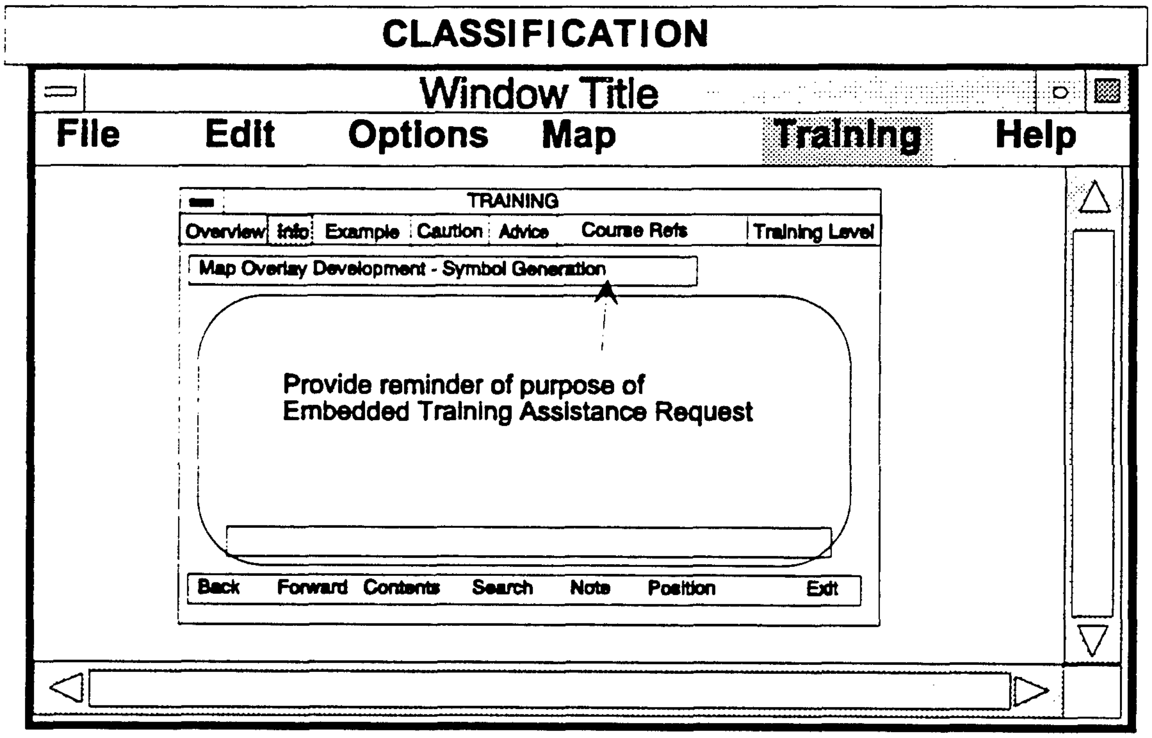
13.5.4 Advance Organization

If the component will be used for knowledge training, provide cues, and overviews that orient users unfamiliar with embedded training content and/or process through brief descriptions of scenarios and exercises or courseware outlines. Stated objectives are an important feature for novice users.

- Provide users a brief statement of the exercise objective. The statement should refer to the primary purpose of the embedded training request.

- Clearly identify each embedded training module. State objective, content, and, where appropriate, the number of subsections and estimated completion time.

- Remind user of the purpose of the request for assistance (see Figure 13-7).



Map Overlay Development - Symbol Generation

Provide reminder of purpose of Embedded Training Assistance Request

Figure 13-7. Example of Assistance Request Reminder

13.5.5 Printing

Provide users the ability to print embedded training content - ranging from screen displays to courseware, information to study further, or for future reference, and/or to print the displayed training material.

13.5.6 Fidelity

Adjust the level of training content and presentation fidelity to match the training:

- Low fidelity for initial training, simple data, and easy process
- High fidelity for unusual processes, hazardous events, or difficult processes.

13.5.7 Simplicity

Give users simple answers to simple questions. If the answer is long or complex, offer a summary and options to request additional guidance.

13.5.8 Verification

Allow users to verify or confirm selected options, solutions, and commands. This allows them to evaluate the completeness of a process or task and the accuracy of their approach without having to sort through extraneous material.

13.6 ACCESSING TRAINING

13.6.1 Displayed Embedded Training Availability

Display the command, icon, or function key used to access training throughout the application to remind the user of training availability.

13.6.2 Access Via Training Icons

Allow users to access the embedded training directly by selecting an embedded training icon and moving to the point of user need. For example, a user could activate embedded training and select a "Cautions" component icon, move to the point where assistance is needed, click, and receive additional information without exiting the application (see Figure 13-8).

13.6.3 Structured Menu

When using a structured menu to access the embedded training, allow users to add to or change existing embedded training messages (e.g., add terms to the menu using an ADD function). If users are allowed to customize menus, the original menu must be protected (e.g., log-on files for individual users).

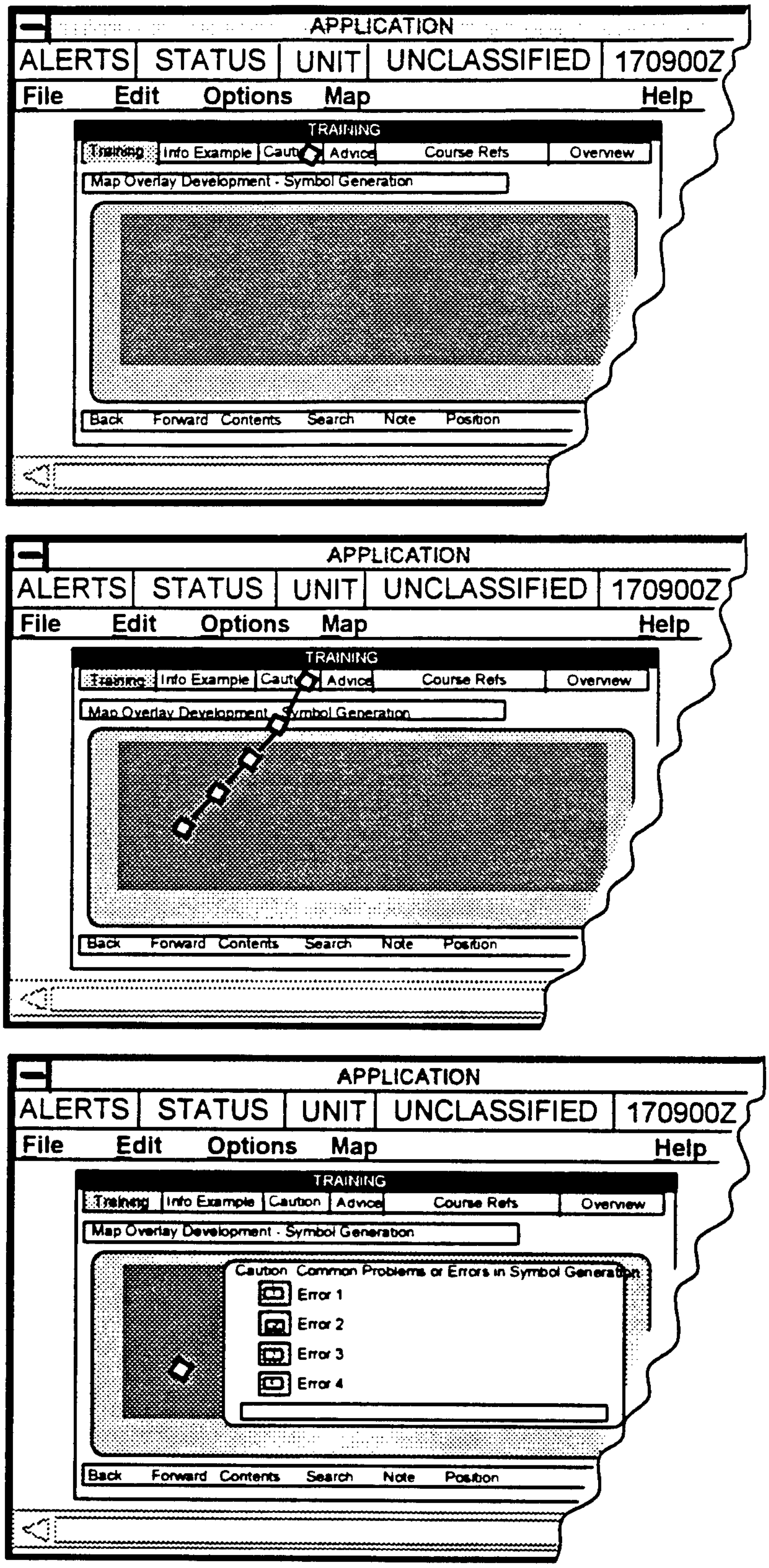


Figure 13-8. Using an Icon to Access Embedded Training Directly

13.7 SCREEN DISPLAY

13.7.1 Complete Display

The content of each screen should stand on its own; do not require users to refer to a previous screen within a module to recall essential information. For example, if users need to enter identical information on a series of screens, the system should automatically enter the appropriate information, repeat the information on each screen, or prompt users to record the information.

13.7.2 Graphics

Select uncomplicated graphics that portray the functional objective clearly, omitting nonessential visual detail.

13.7.3 Window Placement

Display the assistance in windows that do not completely obscure the application's critical navigation buttons, operational icons, or the status message line or window.

13.7.4 User Window Control

Allow users to resize and reposition overlapping windows. This will allow users to see the portions of the application with which they are most concerned.

13.8 TECHNICAL COMMUNICATION/WRITING STYLE

Phrase embedded training topics, messages, and menu options in the active voice. Phrase task-related terms to refer to the learning task (e.g., "Creating and modifying fields" instead of "Fields").

13.9 MOBILITY/NAVIGATION

13.9.1 Mobility Within the Embedded Training

Allow users to move among embedded training components freely:

- Without returning to the top of a central hierarchy
- Without exiting the current embedded training component
- Without having to proceed through a preset path
- Without having to step through introductory material.

13.9.2 Embedded Training Navigation Button Display

Display embedded training navigation buttons in each embedded training screen for controlling movement between and within modules (see Figure 13-9).

13.10 ERROR FEEDBACK

13.10.1 Immediate Feedback

Provide feedback to users in a timely manner, adjusted to the users' expertise.

- When practice requires multiple steps, provide users immediate feedback to avoid a series of incorrect actions.
- For novices and for uncomplicated problems, offer immediate feedback that includes a suggested next best step.

13.10.2 Context-Similarity Feedback

Provide feedback to users in a form similar to the application, product, or outcome (e.g., an error in equipment setup may be illustrated by correctly configured equipment rather than by a checklist, menu, or even natural language message).

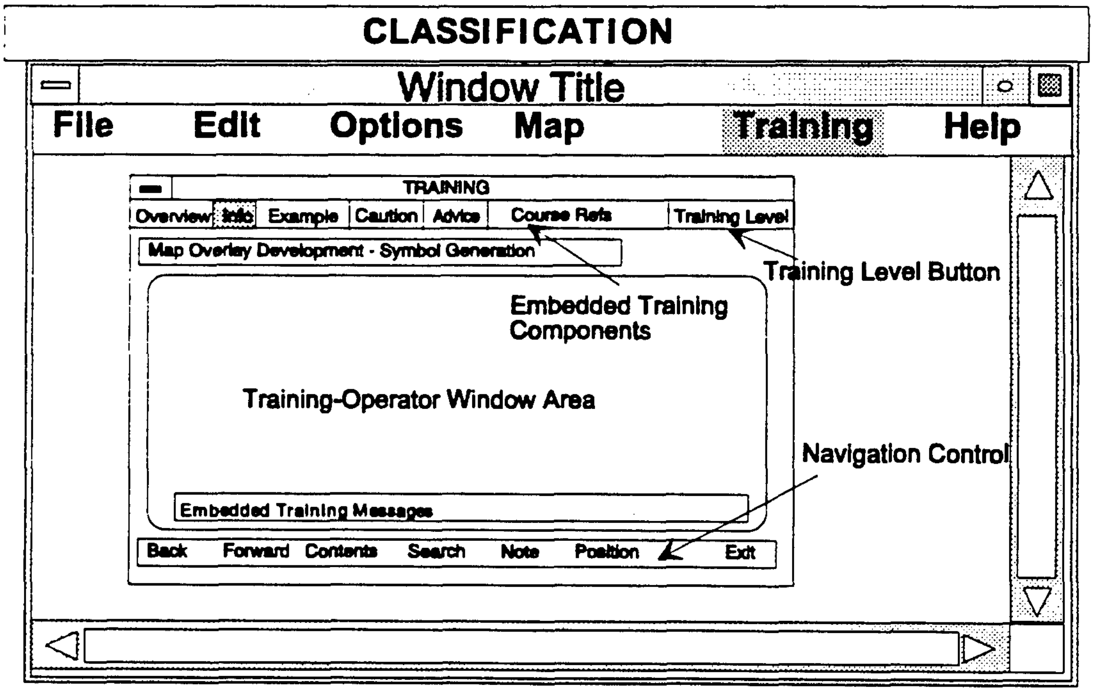


Figure 13-9. Example of Embedded Training Navigation Buttons

13.10.3 Error Identification

Provide specific feedback that identifies errors rather than assigns a score.

13.10.4 Tone of Error Message

Provide error messages that are constructive and neutral in tone; avoid messages that suggest a judgment of the user's behavior. For example, "the system cannot process..." is preferable to, "Invalid Number: Entry must be 4 digits..."

13.10.5 System-Initiated Error Feedback

- When a response to a system-initiated query or recommended action is required, the system should provide users the opportunity to stop and think (i.e., consider other options, recall past experiences, weigh problem solutions) without premature system interruption. For example, the system should avoid over-prompting and unwanted problem resolution. The system could offer users the ability to place the system initiation on hold or cancel an upcoming intervention.

- Provide novice users with prompts identifying probable next-step errors.

- Use control blocking sparingly (e.g., to protect a system from accidental hazardous actuation and system destruction).

- Allow novice users to select an error-blocking option that limits errors.

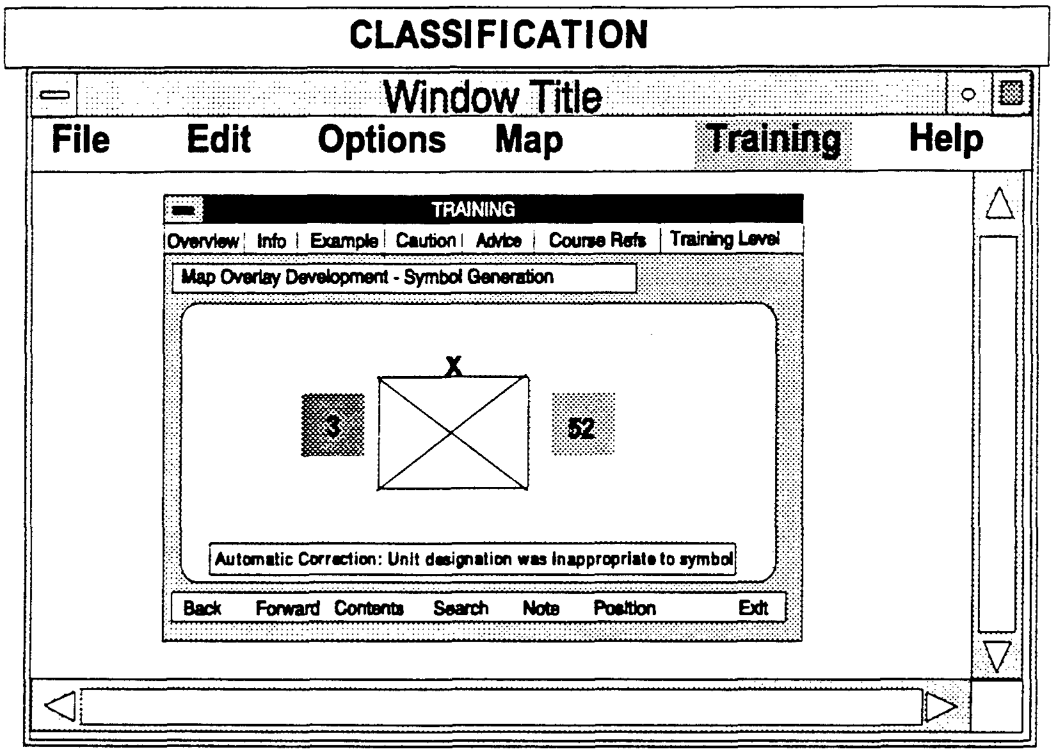
- Give users the option to block temporarily the system-initiated error feedback or instruction.

- If the system automatically corrects some errors (e.g., replacing an out-of-bounds parameter), ensure users are notified of the corrections (e.g., by a message and highlighting of the corrected information) (see Figure 13-10).

- Allow experienced users to select automatic system error correction without requiring their confirmation. This option assumes that mundane errors made by experts are a result of minor actions, such as mis-stiking a key, command, or icon.

- Avoid blocking access to system functionality. This can be very frustrating and can be a result of a misdiagnosed error or correct, but uncommon, approach.

- Avoid user confusion that may result from automatic system error correction.



Map Overlay Development - Symbol Generation

Automatic Correction: Unit designation was inappropriate to symbol

Figure 13-10. Example of Automatic Correction Notification and Identification

13.11 ABILITY TO MODIFY

13.11.1 Additions to the Embedded Training

Provide the capability to add to, but not modify, the original training system. If embedded training allows individual user modifications, protect original application and embedded training (e.g., separate log-on files for each user).

13.11.2 Multi-User Systems

On multi-user systems, permit the individual user to store and reference additions in a individual file.

13.11.3 Annotation

Permit users to annotate a copy of the training program (i.e., examples, pitfalls, process notes, references, etc.).

13.11.4 Annotation Search

Provide users the ability to search an annotation log.

13.11.5 Icons Used to Designate Annotations

Use icons to designate the position of an annotation in the embedded training program (e.g., user example, caution, additional reference) (see Figure 13-11).

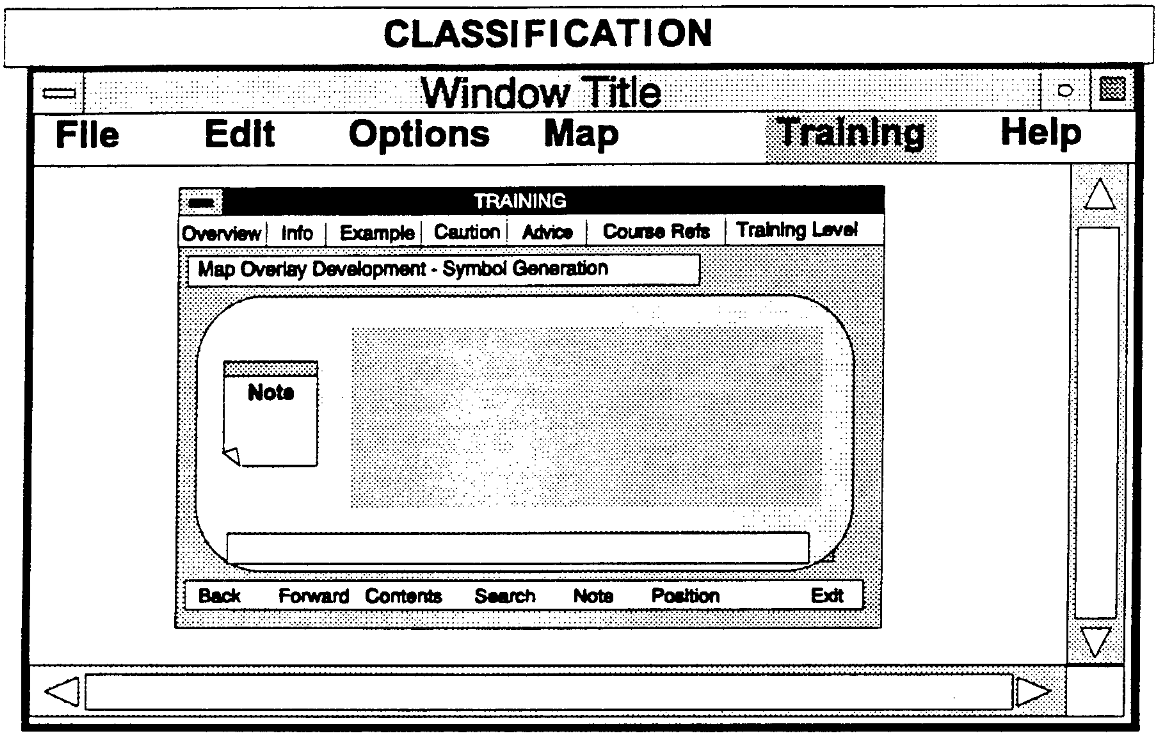


Figure 13-11. Example of Identification Annotation Position

This page intentionally left blank.

REFERENCES

Paragraph	Reference
13.1.1	Nicol (1990) p. 115
13.1.2	Kearsley (1988) p. 27; Shneiderman (1987) pp. 371-372; Carroll and Mazur (1986) p. 38
13.1.3	Shneiderman (1987) p. 374
13.1.4	Kearsley (1988) p. 99
13.1.5	Gery (1991) p. 59
13.1.6	Kearsley (1988) p. 22
13.1.7	Wexelblat (1989) p. 74
13.1.8	Helander (1990) p. 603
13.1.9	Roth et al. (1988) p. 915
13.1.10	Getler (1991) pp.14-22
13.1.11	Roth et al. (1988) p. 98; Carroll and McKendree (1987) p. 24
13.1.12	Avery (1992) Personal Communication
13.1.13	Knerr (1992) Personal Communication
13.1.14	Kearsley (1988) p. 77
13.1.15	Fernandes and Maracle (1991) p. 9; Ripley (1989) pp. 811-822
13.1.16	Roth et al. (1988) p.98
13.2.1	Kearsley (1988) p. 66, 81; Wexelblat (1989) p. 76
13.2.1a	Helander (1990) p. 353
13.2.1b	Kearsley (1988) p. 66, 81
13.2.1c	Kearsley (1988) p. 66, 81
13.2.2	Carroll and McKendree (1987) p. 23
13.3.1	Sellen and Nichol (1990) p. 146; Kearsely (1988) p. 77
13.3.2	Gery (1991) p. 59
13.3.3	Gery (1991) p. 49
13.3.4	Gery (1991) p. 59
13.3.4a	Wexelblat (1989) p. 76

REFERENCES (cont'd)

Paragraph	Reference
13.3.4b	Carroll and McKendree (1987) p. 25
13.3.4c	Carroll and McKendree (1987) p. 25
13.3.4d	Carroll and Mazur (1986) p. 39
13.3.5	Gery (1991) p. 59; Wenger (1987) p. 124
13.3.6	Gery (1991) p. 137
13.3.7a	Seybold's Office Computing Report (1989) p. 6
13.3.7b	Wexelblat (1989) p. 76
13.3.7c	Wexelblat (1989) p. 76
13.4.1	Wenger (1987) p. 336
13.4.2	Fernandes and Maracle (1991) p. 9
13.4.3	Fernandes and Maracle (1991) p. 9; Seybold's Office Computing Report (1989) p. 6
13.5.1	Badler and Webber (1991) p.71; (1990) p. 638
13.5.2	Helander (1990) p. 358
13.5.3	Nicol (1990) p. 115; Roth et al. (1988) p. 118; Carroll and Mazur (1986) p. 38; MIL-STD-1379D (12.5.1990) p. C-5
13.5.4	Carroll and Mazur (1986) p. 37; Walker (1987) pp. 238-243
13.5.5	Kearsley (1988) p. 24
13.5.6	Roth et al. (1988) p. 30
13.5.7	Wexelblat (1989) p. 74
13.5.8	O'Malley et al. (1983) p. 6; Carroll (1982) pp. 49-58
13.6.1	Kearsley (1988) p. 67
13.6.2	Seybold's Office Computing Report (1989) p. 5
13.6.3	Kearsley (1988) p. 79
13.7.1	Fernandes and Maracle (1991) p. 9
13.7.2	Brooks et al. (1990) p. 1387
13.7.3	Fernandes and Maracle (1991) p. 9
13.7.4	Kearsley (1988) p. 76

REFERENCES (cont'd)

Paragraph	Reference
13.8	Sellen and Nicol (1990) p. 145; Helander (1990) p. 360
13.9.1	Fernandes and Maracle (1991) p. 10
13.9.2	Fernandes and Maracle (1991) p. 9
13.10.1	Roth et al. (1988) p. 36
13.10.1a	Roth et al. (1988) p. 36
13.10.1b	Wenger (1987) pp. 296-297
13.10.2	Roth et al. (1988) p. 36
13.10.3	Roth et al. (1988) p. 36
13.10.4	MIL-STD-1472D (1981) p. 274; Shneiderman (1987) p. 317; Smith and Mosier (1986) p. 391
13.10.5a	Wexelblat (1989) p. 76
13.10.5b	Wexelblat (1989) p. 76
13.10.5c	Carroll and McKendree (1987) p. 24
13.10.5d	Carroll and McKendree (1987) p. 24
13.10.5e	Kearsley (1988) pp. 20, 80
13.10.5f	Carroll and Mazur (1986) p. 40
13.10.5g	Carroll and McKendree (1987) p. 24
13.11.1	Kearsley (1988) p. 23
13.11.2	Kearsley (1988) p. 23
13.11.3	Wenger (1987) p. 319
13.11.4	Wenger (1987) p. 319
13.11.5	Gery (1991) p. 63

This page intentionally left blank.

14.0 EMERGING TECHNOLOGY

This section of the *Style Guide* is planned to provide the developer with an overview of new issues that may have an impact on the Human Computer Interface. As design guidelines for these emerging technologies mature, the information presented here may become part of an existing section of the *Style Guide* or form the base for a completely new section in a future edition. The material may be dropped from future versions if no longer relevant. This material includes discussions on personal layers and multimedia technology.

14.1 PERSONAL LAYER

The concept that certain computer interfaces should accommodate different user preferences is widely accepted within the software development community. In a number of situations, this may not be advisable. These situations include multi-user shared workstations, workstations used for over-the-shoulder viewing, and systems that are primarily composed of novice users. In the past and to a certain extent today, the common practice was to assume that there is one "stereotype" user group and to design the interface for that group. Stereotypes (or homogeneous groups) can be defined as user groups formed by individuals with similar or identical characteristics, needs, preferences, and capabilities.

In reality, seemingly homogeneous groups are composed of individuals with widely varying degrees of competence, preferences, and aptitudes. Therefore, in many respects, the system design did not necessarily accommodate these within-group individual differences, with subsequent performance degradation due to:

- Level of experience
- Personality traits
- Demographic characteristics
- Physiological attributes.

Designing a more personalized system that many can use and that remains responsive to individual needs is an elusive goal, primarily because computer-user populations are not homogeneous. Considering individual user differences, it may not be appropriate to design a single static interface. One approach to a personalized interface has been to design different interfaces for different groups of users. The approach requires careful examination of the user population in order to identify different user groups. This may even require different versions of the same product.

Although the need for personalizing the computer interface is generally recognized, the way to accomplish this has not been unanimously accepted. The primary methods or procedures for personalizing an interface include:

- Prototype the application in conjunction with the user.
- Allow the end user to directly modify the working environment.
- Use adaptive modeling.

The first method involves a process by which system designers consult representatives of the end-user population and develop a prototype version of the application. Potential users evaluate general functionality and appearance and comment on system quality. End-user comments are reviewed by system designers and adjustments made to the application. This iterative process continues until the user interface is complete. The end-users involved are assumed to represent the user population as a whole. This is the most common method used by the military operational community. However, a major weakness of this method is the assumption that the combination of individual users constitutes a homogeneous user population.

The second personalization method is to allow the end user to directly modify the working environment. Examples are typically found in the UNIX operating system (discussion follows on UNIX implementations of user preferences). Many researchers agree that the end user should have some ability to modify the interface. For efficiency, techniques are included to allow the experienced user to speed up interactions in natural ways (e.g., enter different information items in the same line to avoid the need for individual prompts [prompt-suppression]). Disadvantages of direct user modification include:

- Difficulty for casual users learning to make modifications
- Having to trade-off between setup time and task to be accomplished
- Difficulty associated with supervisor over-the-shoulder viewing
- Potential between-user difficulties.

The third method of personalization is the adaptive modeling method. Adaptive modeling describes the computer's ability to alter the interface in order to meet the changing needs of the user or to recognize users and adjust based on past preferences or behavior/activities. The system monitors user activity and tries to adapt automatically to the changing behavior.

A "user modeler" is often used to incorporate the user characteristics with other factors affecting performance, preferences, and needs of the user. The purpose of the user modeler is to predict the preferences and the current situation of the user. The user modeler receives data on the user's activities, uses this information together with the profile of the user and a knowledge base already stored on the computer and updates the user model. The model is then used to determine an appropriate interface that fits the user's characteristics, needs, and preferences. The system adapts itself and improves the model as information is collected about the user during the actual interactions.

Some adaptive applications recognize differences between novice and expert users. These interfaces may provide automatic assistance to the novice. However, the expert receives assistance only when it is requested. These applications allow the novice to learn the application more efficiently and to slowly eliminate the tutorial function as application skills improve.

Workgroup situations, such as military tactical operations or business offices, and/or network capabilities in today's workgroups introduce other issues when dealing with personal preferences. For, although it is important that the user have the capability to modify the environment, some order and limitations are necessary. The extent of these limitations depends on their impact. In addition, the challenges of designing groupware -- Computer Supported Cooperative Work (CSCW) -- add a new dimension to the role of interface designer.

Two examples of implementing user preference files can be found in the UNIX operating system and in the WinLogin feature available for the Microsoft Windows operating system. A general outline of each is discussed in the following paragraphs.

Because UNIX is a multiuser system, files can be created by individual users who "own" them until they are deleted or given to another user. Adding a new user to the system requires a user name, group, login identification (ID), and password. Each user belongs to a group and can share files with other members of the group. When a file is created, the user and group are automatically given permission to access the file. The user can add permissions for others or take away the group permission.

In the UNIX system, the user's login ID must be unique to the system. Each user is assigned a "home" directory, which contains the user's personal files. Included in the home directory are the "dot files." The types of preferences specified through dot files include capabilities to:

- Store environment variables
- Store commands that would be typed at the command line
- Create aliases (shorthand forms) for commonly used commands
- Start a window manager (e.g., Motif, Open Look)
- Specify/modify the interface appearance in terms of colors and fonts
- Specify menu items and mouse buttons for selection
- Specify tools and applications available.

The UNIX environment provides the opportunity to customize the working environment, but not without drawbacks. When loading any software applications, environment variables and paths must be set up according to proper specifications. When user environment variables conflict or overwrite software environment variables, the software may not run without making changes. This can make software installations difficult and require the services of system administrators.

Microsoft WinLogin provides a tool for managing workstations on a network running the Microsoft Windows operating system. A user's windows environment is defined by settings in various files (.INI, .PIF, and .BAT) called configuration files. Windows configuration files and configuration files for Windows-based applications are placed at a central location on the network. The network administrator manages the whole set of configuration files as upgrades are made and new applications are added.

A database keeps track of the locations where all configuration files are stored. The network administrator can modify the database using a Configuration Manager. This method facilitates setting up a new application or changing characteristics for groups of users or types of workstations by changing a single configuration file.

WinLogin enables users to log on to any workstation and see their own customized Windows environment. When the user logs in and starts Windows, WinLogin checks the database to locate the user's files, the files for the workstation, and the default settings. These settings are combined based on the merge rules for each database. These merge rules specify that particular entries come from the administrative settings, while others come from workstation and group settings. The merge rules also specify whether "supervisory" entries can be replaced by user preferences.

The following guidelines apply to personalization of the user interface.

14.1.1 Levels of Expertise

- Examine the user group to determine the needs of the individual end user. As a minimum, include expectations of the user's level of experience, personality traits, and demographic and physiological characteristics.

- Provide for the user who is experienced on command line interfaces by allowing for the use of both computer menu sequences and direct commands.

- Provide an adaptive interface design and the capability for the application to interact with the end users on their level of proficiency.

- Determine the standardization requirements of the group (business or operational) before allowing user-controlled interface modification.

14.1.2 Experienced Users

- It is recommended that applications provide program shortcuts for experienced users.

- If direct commands are offered, they should provide a more efficient selection method when proficiency is attained.

- Provide a facility for user-defined abbreviations or aliases (e.g., alias *bye* for *exit* or *logout*).

- Permit the user to specify characteristics of the help system.

14.1.3 Novice Users

Provide the novice user with information and direction. Lead the novice through a solution separate from error messages, which allows the user to call up additional detail. Allow the novice end user minimum options to alter the computer system interface, while allowing the novice user to develop familiarity by using the menu-driven sequences.

14.1.4 Interaction Styles

Design user interfaces uniquely (because individuals are unique) with regard to distinct needs and differences for greater effectiveness. Incorporate the following:

- Ensure that the system is adaptable to the physical, emotional, intellectual, and mental traits of the end-user population.

- Ensure that the system responds to individual differences in interaction manner, depth, and style.

- Utilize user 'stereotypes' in constructing an effective model until preferences are identified.

14.1.5 Interface Personalization

The following principles apply to interface personalization. However, it should be noted that there are situations where personalization is not recommended.

14.1.5.1 Workstations

- Design keeping in mind that differences among users have a greater impact on performance level than differences in system designs and training methods.

- Improve user productivity and efficiency by improving the system's ability to adapt to various user preferences.

- Ensure that the user takes only a minimal amount of time to personalize an interface. If personalizing a system is too complex or requires a considerable amount of time, the effort to personalize will not be cost-effective.

- Enable users to change the appearance of the system interface by changing colors and fonts on the screen, except in circumstances where color is required to be fixed (e.g., security or classification coding).

- Allow users to modify the locations of windows and tool bars.
- Provide options for user manipulation that enable the user to tailor the terminal to his or her individual needs.

14.1.5.2 Networks

- Ensure that network systems allow the user to work in a personalized atmosphere (i.e., users should have the same flexibility that a stand-alone system would offer). To accomplished this, allow each user his or her own network account.
- Allow the user to change the interface so that the same terminal can be attached to different host systems (e.g., make the terminal into a network node by setting the appropriate communications parameters and loading suitable emulators).

14.1.5.3 Messages

- Allow the user to express the same message in more than one way.
- Provide the user with every opportunity to correct his or her own errors.

14.2 MULTIMEDIA

Multimedia blends publishing, entertainment, and computers into a medium for information exchange that expands the potential for all three. Building a multimedia application, often referred to as a "title," requires a mixture of expertise including programmers, writers, artists, musicians, and sound engineers, as well as a multimedia producer to coordinate the activities of the team.

Multimedia elements (e.g., sound, video, animation) are typically sewn together using authoring tools. These software tools are designed to manage multimedia elements and provide user interaction. Interactivity is a main ingredient of multimedia. The user is in control -- what the user sees and hears is the result of choices and decisions the user has made. An important requirement for multimedia applications is that the designer create an interactive environment in which the user is in control and the user is comfortable being in control.

Most authoring tools also offer facilities for creating and editing text and images, and extensions to drive video players, videotape players, and other relevant hardware peripherals. Sound and movies are usually created with editing tools dedicated to these media, and are then imported into the authoring system for playback.

The sum of what gets played back is the human computer interface, and this interface rules both what happens to the user's input and the actual graphics on the screen. Unlike the linear sequence, which defines the order in which the pages of a book are read, a multimedia application allows the user to shift the information focus in a nonsequential manner depending on the reader's interests. The structure of the applications provides options for the reader (e.g., Go to B, C, or D). The author of the text can set up a number of alternatives for readers to explore rather than a single stream of information.

The strengths of multimedia arise from the flexibility in storing and retrieving knowledge. Any information, be it text, graphics, sound, or numerical data, can be linked to any other piece of data, making it possible to create a "seamless information environment."

The hardware and software that govern the limits of what can happen are the multimedia platform and environment. The paragraphs that follow will expand on the various elements that make up the hardware and environment and some emerging guidelines for their use and specification.

Two terms (hypertext and hypermedia) require explanation prior to proceeding.

- **Hypertext** - essentially the ability to link specific text to related text, or in some cases, visual elements. The words, sections, and thoughts are linked together and can be navigated in a nonlinear fashion. Hypertext is an extremely powerful information tool because it allows the representation of knowledge, browsing, carrying out structured searches, and making inferences, all within the same environment.

- **Hypermedia** - when associated images, video clips, sounds, and other exhibits are added to the hypertext. Also, when interaction and cross-linking are added to multimedia and the navigation system is nonlinear, multimedia becomes hypermedia.

Most of the information in the following paragraphs is true of hypertext and hypermedia. These two terms are often used interchangeably in the literature.

14.2.1 Multimedia Personal Computer (MPC)

The Multimedia Personal Computer (MPC) Marketing Council's MPC specification requires that machines bearing the "MPC" trademark offer a core set of features. The specification has been divided into Level 1 and Level 2. The Level 2 specification is recommended. It is also recommended that the hardware acquired be the most affordable in each feature category. Make sure the system allows room for expansion.

A number of companies offer fully integrated Multimedia PC hardware systems. Alternately, upgrade kits are available to transform a current PC hardware (80286 up) into a Multimedia PC. The kits usually contain a sound card with Musical Instrument Digital Interface (MIDI), Compact Disk-Read Only Memory (CD-ROM) drive, and Multimedia Windows software.

14.2.1.1 CPU

Hardware equipped with an 80286 or compatible processor chip is required for Level 1. Generally, at least a 80386 or 80486SX processor or equivalent is recommended; the highest affordable clock speed is desirable. The Level 2 specification requires a 486SX-25.

14.2.1.2 RAM (Random Access Memory)

A minimum of 2 megabytes (MB) RAM is required by MPC Level 1 specification and 4 MB for Level 2. At least 8 MB is recommended for authoring and at least 6 MB for a system that will be used as a presenter only. All systems potentially benefit from more RAM.

14.2.1.3 Magnetic Storage

- A 3.5" floppy drive with 1.44 MB capacity is required.
- A hard drive of at least 30 MB is required for Level 1 and 160 MB for Level 2, but 300-600 MB is recommended.
- A tape drive and tape backup are recommended.

14.2.1.4 Optical Storage

CD-ROM, with compact disc (CD) digital audio output and data transfer rate of 150 kilobytes (kB) per second, is required for the Level 1 specification MPC standard, but 300 kB is required for Level 2. The fastest transfer rate available (at an affordable cost) is recommended.

14.2.1.5 Audio

The following audio hardware is required for MPC:

- An 8-bit for Level 1 and 16-bit for Level 2 digital-to-analog converter (DAC) 22.05 and 11.025 kilohertz (kHz) rate
- An 8-bit for Level 1 and 16-bit for Level 2 analog-to-digital converter (ADC) 11.05 kHz rate, microphone level input
- A music synthesizer capable of four to nine instrument synthesis
- An on-board analog audio mixing capability.

14.2.1.6 Video

VGA (16 colors) color graphics adapter is required, but Super Video Graphic Adapter (SVGA) (256 colors) is recommended. This is still inadequate for producing realistic images or video.

Higher than 640 x 480 resolution is only important when the display is larger than 14". On a 14" display at a normal desktop viewing distance of about 18", the user cannot distinguish more than 640 pixels across. When using a 16" or 19" display at this viewing distance, an Extended Graphic Adapter (EGA) display of 1024 x 768 can be beneficial.

14.2.1.7 I/O Hardware

The following I/O hardware is required for MPC:

- An 101-key keyboard
- A two-button mouse (three-button is acceptable)
- A MIDI I/O port
- A serial port
- A parallel port
- A joystick port.

14.2.2 Audio

Audio is an integral part of the multimedia environment adding the dimensions of speech, music, and/or sound effects. The ability to capture natural sounds and bring them into a multimedia application is the purpose of digitized audio. This involves more than setting up a microphone and telling the computer to capture the sound. A well-produced sound track is the best means to enhance the realism and effectiveness of the application interface. The following paragraphs discuss the facilities necessary to capture and process audio.

14.2.2.1 Audio Digitizers

Audio digitizers are devices for recording and playing back digital audio. They range from the simple add-in sound cards to the audio production systems costing thousands of dollars. The principal technical descriptive characteristics consist of sampling rate, sampling size, and resolution. The selection decision should be based on a trade-off analysis of user functional needs and equipment availability.

- Sampling Rate: The sampling rate is like the frame rate at which film or video is played back. Sounds sampling or digitizing captures "snapshots," samples of sounds that are played back rapidly. The MPC specification requires mono playback at 22.05 kHz but recording at a rate of 11.025 kHz. Most PC sound boards allow playback and recording at 44.1 kHz (compact disc audio plays back at a rate of 44.1 kHz).

- Sample Size: The amount of information stored about each sample. Sample sizes are typically either 8-bit or 16-bit. The larger the sample size, the better the data describes the recorded sound. While 8-bit sound provides 256 units to describe dynamic range and amplitude, 16-bit provides 65,536 units.

- Resolution: The resolution is the number of bits used to represent an individual sample. The more frequent the sample and the more data stored about the sample, the finer the resolution and quality of the captured sound when it is played back. It is analogous to the number of bits used to represent a pixel on a screen. As an 8-bit image is grainier than a 16-bit image, an 8-bit sound is grainier than a 16-bit sound. The MPC standard supports the 8-bit rate, but the 16-bit rate is recommended for quality sound reproduction.

14.2.2.2 Sound Editors

Sound editors (also called sample editors) provide tools to record, edit, rearrange, mix, process, and playback sound files in a variety of formats. Sound is represented as an amplitude waveform, with time corresponding to the horizontal axis and sample value (i.e., volume or intensity) assigned to the vertical axis. Within this format, the user can zoom out to view an entire sound file or zoom in as close as a single sample. This allows cutting, copying, and pasting of sound data with a precision of up to 1/44,100 of a second, depending on the digital signal processor (DSP) board and software.

Sound editors are available both as stand-alone products and bundled with digital audio cards. Most bundled editors include a limited number of features such as cut, paste, fade-in and out, and amplitude adjustment.

Full-fledged editors provide signal processing options such as cross-fading, which allows one track to fade in while another fades out; digital equalization features, such as boosting or cutting the volume of selected frequencies or frequency bands; time compression and expansion, increasing or decreasing the length of a sound file region without changing its pitch; and pitch shifting, changing the key of a passage without altering its duration. These options are invaluable for fitting a piece of audio to video not specifically created for the sound or sound not created for video.

14.2.2.3 MIDI

MIDI is an international specification used by electronic musical instruments to communicate with each other, computers, mixers, and other devices. MIDI specifies the cabling, hardware to connect MIDI devices, as well as protocol to communicate between these devices. Any musical instrument with a microprocessor to process MIDI messages can be a MIDI device.

Whereas digital audio actually records and stores the sound, MIDI simply describes the performance. MIDI describes what notes are being played, when they are played, and with what nuance (e.g., sustain, pitchbends, vibrato). Since MIDI deals only with events that trigger sound, the files are rather small (e.g., a one-minute sampled composition requires 12 MB to store but only 15 kB MIDI file). Playing back a MIDI file requires a musical instrument (which could consist of a box and speaker outputs), while the sound recording requires only an amplifier and speakers.

- IBM- and MPC-compliant sound cards usually include basic MIDI interfaces.

- Dedicated interface products are more appropriate for musicians and multimedia experts. They often allow increasing the available channels and therefore allow controlling more devices.

14.2.2.4 MIDI Sequencers

MIDI sequencers record and store musical events played on a MIDI instrument such as a synthesizer or sampler. They do not record the sound but record the MIDI data describing the sounds. Most MIDI sequencing software mimics a typical multitrack tape recorder and offers standard editing features such as cut, copy, paste, merge, insert, pitch correction, transposition, inversion, retrograde, tempo changes, and score edits.

A MIDI file can contain up to 16 channels of music data, allowing recording of and playing back of many different musical instruments, each on a different channel. The general MIDI numbering system from 0 to 127 identifies instruments that can be synthesized, although MIDI is flexible enough to allow remapping to non-standard instruments. MIDI also allows the user to set up any instrument to receive on any channel (e.g., data could be received from a keyboard synthesizer and played back on a another keyboard synthesizer or MIDI instrument).

14.2.3 Images

Software is available to support nearly every combination of 3-D modeling, lighting, defining surface attributes, animating, and rendering. Selecting the appropriate software depends on the type of graphics. Print work, animation, visualization, fly-throughs, slides, and multimedia all require different subsets of features.

The computer creates still images in two ways -- as bitmaps (or paint graphics) and as vector-drawn (or drawn) graphics. Bitmaps are used for photo-realistic images and for complex drawings requiring finer detail. Vector-dawn objects are used for lines, boxes, circles, polygons, and other graphic shapes that can be expressed in angles, coordinates, and distances. A drawn object can then be filled with color and patterns.

The appearance of these graphics depends on the display resolution and capabilities of the computer's graphics hardware and monitor. The images are stored in various file formats and can be translated from one application to another and from one computer platform to another. They are typically compressed to save memory and disk space.

Programs are available for converting between the two formats. Converting a drawn (or vector) object to a bitmap is far easier than the reverse.

14.2.3.1 Painting Programs

Clip Art is an example of commercially available bitmapped graphics. Clip Art can be manipulated, and properties such as brightness, contrast, color depth, hue, and size can be adjusted.

Bitmap editors are usually called paint programs. They are the closest to the traditional artist's media and creative processes. They determine the color of each pixel they touch. The tools shape and shade the images by manipulating brush type, geometry, and ink style. The greater the control, the greater the number of effects that can be created.

Fill tools place solid colors, textures, and patterns in the designated areas. Areas can also be selected for cutting, copying, and pasting operations as well as rotating and scaling.

Considerations in purchasing a painting tool include the capabilities of the toolkit, maximum image size and resolution, interface resolution, and the number of colors available.

14.2.3.2 Drawing Packages

Computer-Aided Design (CAD) programs traditionally use vector-drawn graphics for creating the highly complex and geometric renderings needed by architects and engineers. Programs for 3-D animation also use drawn graphics.

Vector graphics software are called draw programs. These tools are similar to those used in mechanical drawing. They generally include tools for creating geometric shapes, lines, and curves. Objects can be moved, scaled, rotated, copied, and attributes changed.

14.2.3.3 Animation

The entire project can be animated, or animation can be used for accenting. Animation can consist of as many as 30 images per second. One way to generate animation is to create a series of still images individually and use animation software to flip through them like a movie.

The capability to import and integrate images from a wide variety of sources is an important feature. Fast rendering, timeline- and keyboard-based animation interfaces, and 3-D fonts are features of a higher end tool.

14.2.4 Video

Video differs from animation in that video describes images of real events stored in a digital format, whereas animation is simply computer-generated images. Video image files usually contain audio tracks and are larger than animated images.

Although many applications are created using images or animation or both, video appears to have the greatest user impact. Video must be well planned to have the greatest impact and effectiveness. Integration of video into the application and disk storage space are key elements for successful video.

Two new products for video incorporation using only software support are described, followed by a discussion of hardware-based tools.

14.2.4.1 QuickTime Movies

Many Apple-Macintosh applications can create or play QuickTime movies. QuickTime for Windows (QTW) allows PC users to access all of the QuickTime movies available for the Mac. This enables the same video clips to be used on both platforms.

Dedicated editing and effects software is the best choice for polishing productions. Just about anything you can do in a broadcast-quality editing suite has a QuickTime equivalent, limited only by today's lower hardware resolution and size and frame rates. Editing functions include "log," mark, and identify scenes or picture sequences; trim to desired length; and order them for playback. Most editors have a visual interface to identify and sort clips and a timeline view for sequencing and trimming elements.

Some editors allow creation of transitions, adding titles and graphics and applying various image transformations and filters such as traditional wipes, flips, and turns as well as digital domain unique morphs and melts. The transitions and overlays available are dependent on the special effects capabilities of the system to translate perfectly.

Audio support is often limited to capturing audio on suitably equipped computers, trimming the segments and adjustment of audio and visual segments. For the full range of audio effects, a sound editor is necessary.

14.2.4.2 Video for Windows

Video for Windows (VFW) is also known as Audio Video Interleaved (AVI). It allows the developer to capture, digitize, and compress (using a number of different compression algorithms) video. Because of the software-only-compression, compromises must be made. The image size is small, and interleaving is required to synchronize audio and video.

VFW is scalable; it can be played back on the user's PC, with an additional video decompression board. The quality of the video being played back depends on the power of the playback PC. The software drops frames when necessary, to ensure that the audio stays synchronized to the video sequences.

On a slow 386, the video may play back at 10 frames per second (fps). On a fast 486, the playback can be at 24 fps. The slower 10 fps rate will produce low quality video with a large amount of flicker.

VFW is installed as a multimedia device.

14.2.4.3 Video Capture

Capturing still images from video segments is difficult; grabbing complete sequences of images truly taxes the current capabilities of the system. Uncompressed, full-motion video is impractical because 30 seconds of full-motion video stored in analog form requires over 500 MB of storage. To make video manageable, the file must be compressed.

A growing number of manufacturers are offering add-on boards, called frame-grabbers, that grab and store movie-video images in digital form. Video capture boards can be distinguished by whether they convert full-screen, full-motion video [30 frames per second in the U.S. - National Television Systems Committee (NTSC) standard, or 25 frames per second in the European Phase Alteration Line (PAL) standard] or by whether they capture selected frames or partial screens.

The number of colors varies. Most designs that started in the video world grab 16 bits per pixel, while most computer-oriented products capture 24 bits per pixel. Few boards can grab a complete video signal at full speed.

- Applications include traditional video editing, multimedia presentations, training videos, kiosks, scientific analyses, and archival storage.
- Some are offered as separate boards that accept analog video as inputs and produce digital files as output.
- More often, digitizers are combined with video capabilities in a single board in a motherboard-daughterboard configuration or as side-by-side boards connected by a ribbon cable.

14.2.4.4 Compression

Many products trade off a higher frame rate for lower pixel depth or smaller image size. Boards that capture images at rates approaching full speed can normally do so only to RAM, since data cannot be written to a hard disk fast enough.

The current generation of PCs requires hardware-assisted compression to capture full-screen, full-motion video. Most boards have relied on either Intel's Digital Video Interleaved (DVI) chip sets or on C-Cube CL- 550 Joint Photographic Experts Group (JPEG) processor. Since video compression standards have been in flux, companies have tended to use various methods. Systems based on the JPEG are the most prevalent, while DVI and Motion Picture Expert Group (MPEG) are gaining popularity. A brief discussion of these three standards is presented below.

- Under JPEG, an image is divided into 8 x 8 pixel blocks, and the resulting 64 pixels (called a search range) are mathematically described relative to the characteristic of the pixel in the top-left pixel. Since the binary description of this relationship requires less than 64 pixels, more information can be transmitted in less time. JPEG is primarily used to encode still images and compresses about 20:1 before image degradation occurs. Compression is slow. JPEG does not handle black and white (1-bit per pixel) images.
- MPEG is used to encode motion images. MPEG compresses at a 50:1 ratio before degradation of the image occurs. Ratios of 200:1 are attainable, but observable degradation occurs. The compression rate is fast enough to allow CD players to play full-motion color movies at 30 frames per second.

- DVI is a proprietary, programmable compression/decompression technology. The hardware consists of two components to separate the image processing and display functions. It allows compression of video images at ratios between 80:1 and 160:1. DVI will play back video in full frame size and in full color at 30 fps. When tied in with a mainframe computer, DVI playback approaches the quality of broadcast video.

Although DVI claims to offer greater compression ratios, JPEG independently codes each frame. This allows frames to be edited or rearranged. Faster DSPs or faster CPUs may eventually provide new compression methods.

Boards without dedicated chips can compress video captured into memory. A PC or Mac with 8 MB of RAM can usually capture a few seconds of partial- frame video before stopping to compress and save to disk.

Most applications can use less costly partial-screen or slower-frame-rate videos. These applications (e.g., electronic mail or training) often limit video to partial screen in order to provide room for other program elements.

Systems designed to play back QuickTime movies on the Mac and PC or AVI on the PC are limited by the playback hardware. Most Macs and PCs are limited to a partial screen and about 15 frames per second.

14.2.4.5 Video-Editing Software

Desktop video-editing systems vary widely in capability to handle video. Less expensive packages deal mainly with control and status information, while the actual video signals are routed directly from recorder to recorder or recorder to screen. The packages are classified as follows:

- **Cuts-only** - unadorned final output is copied directly from one deck to another

- **Off-line** - an edit decision list (EDL) will be exported to a more sophisticated editing system

- **On-line** - can add graphics, transitions and special effects to video. On-line systems can also produce EDLs for further work on larger systems.

Generally, compressed video is good enough for EDL; but the amount of compression required to get the original video down to practical sizes squeezes out a fair amount of the picture quality. Most digital systems are used as off-line feeders to an on-line system. However, as compression technology advances and storage options become less expensive, more digital systems will offer direct output alternatives.

Older desktop video editing systems are similar to traditional editing systems. The newer systems have graphical interfaces using a point-and-click operation. The intended audience often dictates the capabilities required. The ability to connect to video decks and recorders through distributed control networks allows editing of specific frames.

14.2.5 Text

Many multimedia applications are primarily text-driven. Text-based files form one of the largest sources of information. Often multimedia projects are developed by converting a book into an on-line application. The three main ways to get text into compatible forms include the following.

14.2.5.1 Retyping

Although retyping the text can be labor-intensive, it is often the most economic way to import large amounts of printed material.

14.2.5.2 Scanning

Scanning can be an efficient way to get text into a computer. The scanner converts pages of text into bitmapped images. Software is used to analyze the letter shapes and convert them to ASCII letters. Utility programs are available to detect misreads and scanner errors.

14.2.5.3 Computer-Supported Conversions

Converting involves transferring electronic files between different formats. Converting text always results in the loss of some original formatting. Proper formatting, indexing, and other reference tags are required to make the text useful. Suggested formats include straight American Standard Code for Information Interchange (ASCII) text, Rich Text Format (RTF), Standard Generalized Markup Language (SGML), and Document Control Architecture (DCA).

14.2.6 Compact Disc Technology

CD technology is gaining acceptance as an economical storage medium, which is ideal for delivering large programs such as reference material and multimedia titles. PCs, Macs, and workstations are now available with CD-ROM drives and upgrade kits. Drives are becoming less expensive, and consumers and education audiences have increased expectations.

14.2.6.1 CD Specifications

MPC Marketing Council has stated that CD-ROM drives must be able to read multisession recordings and be CD-ROM eXtended Architecture- (XA) ready. The XA files produced by the Kodak Photo CD format is an example of ready-to-read XA files. Competing CD formats include Commodore's Dynamic Total Vision (CD-TV), Sony-Phillips' Compact Disc - Interactive (CD-I), Tandy's Video Information System (VIS), Sony's Multimedia CD Player (MMCD), and Kodak's Photo CD.

14.2.6.2 Storage Capacity

A single CD-ROM disc can hold up to 680 MB of information. This equates to 150,000 printed pages or approximately 250 large books on one compact disc.

14.2.6.3 Data Transfer Rate

Data transfer rates have increased from 150 kB per second to 300 kB per second. Increased speed allows smoother audio and video playback. Although new 16-bit game cartridges are better, only CD-ROM has the potential to truly increase games productions, which will include lengthy stereo audio and full-motion video clips.

14.2.6.4 Access Time

The average access time is the time it takes to find what you want to read from the disc. Average access times (also called seek time) for state-of-the-art systems is about 280 ms, although most drives are still in the 350-380 ms range. The MPC standard is anything under 1000 ms, as opposed to 15-30 ms for most contemporary hard drives.

The objective in designing a multimedia interface is to reduce the number of seeks the drive makes in order to access the data. Advanced CD mastering packages allow selection of the ring where data are located.

14.2.6.5 Mastering the Title

The process of turning an application and its associated files into a CD-ROM disc includes premastering, final testing, and mastering and replication.

14.2.7 Authoring Systems

Multimedia elements are typically sewn together using authoring tools. These tools provide the basic building blocks and framework for creating a multimedia application. In designing the multimedia project, the traditional scripting and design methods (e.g., copyboard drawings and typed scripts) or software tools can be selected.

Most authoring tools can be used by non-programmers, although some programming is useful. Authoring tools are best suited for content-rich applications (e.g., those loaded with text, images, and sound) because they specialize in data delivery. Their benefits include ease of use, fast development cycles, predictable characteristics, and reliability.

The target audience is the most important factor in selecting an authoring system. Developing a package for in-house use or controlled situations (e.g., kiosks) provides more leeway than productions for a demo or a tutorial for distribution to a large audience.

Creating a presentation capable of running consistently on almost any machine requires an authoring system suited to the purpose. Multimedia desktop presentation packages vary from pure 2-D animation programs to traditional charting packages with a few added multimedia capabilities. These evolving presentation packages may be the best choice for users already familiar with charts and slide shows.

Features of a good authoring system include:

- Capability to integrate text, still graphics, animations, sound (digitized, MIDI or CD-Audio), and video
- A visual flowcharting system, storyboards, navigation diagrams, or overview facility for illustrating the project structure at a macro level
- Support for creating tailored presentations, either through scripting language or other means (often icon-based programming)
- Support for one or more levels of interactivity
- Capability to allow specification of timing and sequence on systems with different (faster or slower) processors
- Provision of a playback feature for building and testing segments of the project
- Permission to distribute run-time files created
- Capability to add new features or extensions.

Many of the newer packages take a middle-of-the-road approach by combining text, graphics, sound, video, and 2-D animation. Other packages contain tools to create objects and other tools to animate them. As the artistic power and complexities increase, so does the time and training required to master them.

14.2.7.1 Types of Authoring Tools

Various authoring tools can be grouped based on the concepts used to sequence and organize multimedia elements and events. In choosing an authoring tool, the developer must ensure the tool supports the types of things the application will do. The various groupings are discussed below:

- **Card- or Page-Based Tools** - Elements are organized as pages of a book or stack of cards. The pages or cards can be linked in an organized sequence. The user can jump, on command, to any card in the sequence.
- **Icon-Based Tools** - Multimedia elements and interaction cues (events) are organized as objects in a structural framework or process. They simplify project organization and display flow diagrams of activities along branching paths.

- **Time-Based Tools** - Elements and events are organized along a time line, with resolutions as high as 1/30 second. They are best when the project's message has a beginning and an end. The sequentially organized frames are played back at the speed set and other elements triggered at a given time or location. More powerful tools allow jumps to any location in the sequence. These jumps provide navigation and interactive control.

14.2.7.2 Multimedia Integration Tools

Multimedia brings together data from a variety of sources. The authoring package should allow importing files created in a variety of formats, including graphics, animations, sound, and text. Text should be capable of being imported with format intact (see Paragraph 14.2.5). Software should allow integrating graphics files created in other programs with minimal editing or use of specialized conversion packages.

Multimedia integration tools fall somewhere between presentation and animation packages. They have a lot in common with authoring packages but do not usually include the scripting languages of authoring systems. Some features include:

- More control of the various media than with presentation packages
- Allow capture or import graphics and text from other programs
- Provide the ability to put object in motion and synchronize that motion to sound and video clips
- Provide support for multimedia peripherals.

14.2.7.3 Platform

Platform considerations for the authoring systems include the following:

- **Personal Computers** - If the authoring system runs in Windows, although this will assure the presentation runs in Windows, remember that there are several types of Windows. While many features of Multimedia Extensions for Windows are built into Windows 3.1, some are not. A DOS-based product will provide access to the widest possible audience. Ensure that the system supports the wide range of video adapters, memory- addressing schemes, and other features available on DOS machines. Note that not all DOS machines have a mouse, but MPC-compliant machines will.
- **Macintosh** - Ensure that the platform matches the color and software requirements (e.g., System 7 or QuickTime) specified by the authoring package. Requiring such devices as CD-ROM drives, laserdisc players, or MIDI interfaces may require separate routines.
- **UNIX** - Few commercial tools are currently available.

- **Cost** - Prices for authoring systems vary widely. Learning curves and technical support should also be considered when deciding on an authoring system. Some vendors include training as part of the product package.

Consider also the costs of distributing presentations (e.g., runtime versions and unlimited distribution licenses). Some include no-cost runtime licenses, while others adjust their price according to the number to copies you want to distribute.

14.2.8 Design Guidelines

The user interface is the portion of the multimedia application that presents the choices and requests to the user, receives input from the user, and provides feedback about status. The designer should not assume any knowledge on the part of the user. All information about what to do next should be constantly available on screen or in audio.

14.2.8.1 User Task Analysis

- The multimedia interface designer must have a full appreciation of how the user will use the application and what is expected from the application.
- Develop a road map of how the user will proceed through the application.
- Use an expert on the textual content for determining what topics need to be related, how available information should be divided into digestible topics, and the order in which the topics should be presented.

14.2.8.2 Novice vs. Expert Interface

- Provide plenty of navigation power, access to content and tasks for users at all levels, and a HELP system for reassurance. A separate interface for users at different experience levels is not necessarily the best method in multimedia projects.
- Present all information in easy-to-understand structures and concepts; use clear textual clues.
- The best user interface requires the least learning effort.

14.2.8.3 Design Consistency

- Ensure consistent internal design (e.g., topic screens look alike, type faces are consistently used).
- Integrate all elements, such as audio, graphics, and animation, cleanly into the overall feel of the application.
- Take time to identify some basic design standards at the beginning of the design process.

- Decide on the overall concept or metaphor and ensure all elements build on and reinforce the metaphor.
- Ensure consistent style in terms of content and breadth of information.

14.2.8.4 Data Types

- A good multimedia application will include only a fraction of the many data types that are possible.
- Use visual tools to enhance information retrieval. Do not confuse or distract the user with visual elements used simply because they are available.
- Limiting the data types also simplifies the installation of applications on different systems.

14.2.8.5 Navigation

- An important requirement for multimedia applications is that the designer create an interactive environment in which the user is in control and the user is comfortable being in control.
- The application should allow users to start when they want, stop when they want, retrace their steps when they want to backup, and, most important, never do something they don't expect.
- Provide the user the sense of freedom of choice, but remember too much freedom can be disconcerting and users may get lost.
- Provide an escape path for the user in every part of the application. The controls for the escape should be on-screen.
- Try to keep messages and content organized along a steady stream of major subjects while allowing the user to branch outward to explore details.
- The structure can be designed as a linear sequence of chronological events but also allow jumping to a specific event.

14.2.8.6 Cross-Reference Jumps

The most common multimedia behavior is a response to clicking on active words in the client area of the application. Clicking on certain text strings causes a new linked screen of information to appear. These text strings are referred to as "hotwords" or "hotspots." The result of clicking on a hotword is a jump. Picture hotspots work in the same manner. Most authoring tools provide a means to connect related topics through cross-reference jumps.

Buttons, hotwords, and picture hotspots are the primary means of control in multimedia applications. When designing cross-reference jumps, remember:

- Large numbers of jumps make navigation complex and increase test time.
- Ensure all jumps serve a useful purpose.
- Jumps should only occur between directly related or equivalent topics.

Provide an easy way out of the side trip. The user should never feel penalized for exploring (e.g., escape path discussed in Paragraph 14.2.8.5d).

REFERENCES

Paragraph	References
14.1	Aykin and Aykin (1991); Egan (1988); Elkerton and Williges (1989); Fowler, Macaulay, and Fowler (1985); Greenberg and Witten (1985); Hansen (1971); Innocent (1982); Rich (1983); Shneiderman (1982)
14.1.1a	Egan (1988) p. 543; Aykin and Aykin (1991) pp. 373-374
14.1.1b	Elkerton and Williges (1989)
14.1.1c	Greenberg and Witten (1985) pp. 31-33
14.1.1d	Aykin and Aykin (1991) p. 373
14.1.3	Greenberg and Witten (1985) p. 32; Elkerton and Williges (1989)
14.1.4	Aykin and Aykin (1991) pp. 373-374
14.1.5.1a	Egan (1988) p. 543
14.1.5.1b	Greenberg and Witten (1985) p. 31-33
14.1.5.1c	Greenberg and Witten (1985) p. 31-33; Elkerton and Williges (1989)
14.2	Bunnell (1993); Microsoft (1991b); Vaughan (1993); Rosenborg (1993); Burger (1993); Luther (1992); Wodaski (1992); Nielsen (1990); Shneiderman and Kearsley (1989); Seyer (1991)
14.2.1	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3; Flynn (1993) p. 30
14.2.1.1	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3; Vaughan (1993) p. 169
14.2.1.2	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3; Vaughan (1993) p. 171
14.2.1.3	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3
14.2.1.4	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3; Vaughan (1993) p. 188
14.2.1.5	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3
14.2.1.6	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3; Rosenborg (1993) p. 420-422; Burger (1993) p. 164-172; Luther (1992) p. 19-20
14.2.1.7	Bunnell (1993) p. 51; Microsoft (1991b) p. 1-3

REFERENCES (cont'd)

Paragraph	References
14.2.2	Luther (1992) p. 163
14.2.2.1	Bunnell (1993) p. 5; Burger (1993) p. 35-38; Wodaski (1992) p. 59-60; Rosenborg (1993) p. 412-413; Vaughan (1993) p. 41-42
14.2.2.2	Bunnell (1993) p. 9
14.2.2.3	Bunnell (1993) p. 14
14.2.2.4	Vaughan (1993) p. 44-51
14.2.3	Bunnell (1993) p. 57; Vaughan (1993) p. 60-68; 74-77; Rosenborg (1993) p. 39-41; 425; Burger (1993) p. 174-178
14.2.4	Rosenborg (1993) p. 44-47
14.2.4.1	Bunnell (1993) p. 71; Vaughan (1993) p. 279-282
14.2.4.2	Vaughan (1993) p. 279-282; Rosenborg (1993) p. 46; Wodaski (1992) p. 178-179
14.2.4.3	Bunnell (1993) p. 73
14.2.4.4	Bunnell (1993) p. 73; Vaughan (1993) p. 414-416
14.2.4.5	Bunnell (1993) p. 77
14.2.5	Microsoft (1991b) p. 8-1 through 8-11
14.2.6	Bunnell (1993) p. 39; Wodaski (1992) p. 171; Microsoft (1991b) p. 11-3 and pp. 4-4 through 4-5; Vaughan (1993) p. 424-425; Flynn (1993) p. 30
14.2.7	Bunnell (1993) p. 27; Vaughan (1993) pp. 5, 141, 217, 220, and 222; Microsoft (1991b) pp. 2-6, 2-12
14.2.7.1	Vaughan (1993) p. 218-219
14.2.7.2	Bunnell (1993) p. 27
14.2.7.3	Bunnell (1993) p. 27
14.2.8	Luther (1992) pp. 108-109
14.2.8.1	Luther (1992) p. 109; Microsoft (1991b) pp. 2-7 to 2-9; Rosenborg (1993) p. 383
14.2.8.2	Vaughan (1993) pp. 138-139
14.2.8.3	Luther (1992) p. 110; Microsoft (1991b) p. 2-9

REFERENCES (cont'd)

Paragraph	References
14.2.8.4	Rosenborg (1993) pp. 390-391
14.2.8.5	Luther (1992) pp. 11-12; Vaughan (1993) pp. 141-143; Rosenborg pp. 24-27 and 240-242
14.2.8.6	Microsoft (1991b) p. 2-8; Rosenborg (1993) pp. 240-242

This page intentionally left blank.

APPENDIX A

SECURITY PRESENTATION GUIDELINES

This page intentionally left blank.

APPENDIX A

SECURITY PRESENTATION GUIDELINES

This appendix seeks to provide a uniform HCI across all CMW applications.

The security portion of the HCI will comply with DDS-2600-6216-89 and the DIA Style Guide, from which this appendix is derived. These documents outline security-related interface requirements for workstations operating in the System High or Compartmented Mode. To ensure consistency, however, any DoD workstation security label displayed by an application should conform to the labeling guidelines in this appendix.

> (NOTE: *Although the figures in this appendix are only drawn in the Motif style, the security relevant information is common to both Motif and Open Look applications.)*

A.1 LABEL STANDARDIZATION

The following guidelines for label presentation apply to all CMWs.

A.1.1 Guidelines for Label Syntax

One of the primary display requirements is to use the long names of words in all labels.

A.1.1.1 Sensitivity Labels

- The syntax for output of sensitivity labels for all CMWs is as follows:

  CLASSIFICATION COMPARTMENT COMPARTMENT COMPARTMENT....
  Examples: TS A
  TS A B C D

- The syntax for input of sensitivity labels for all CMWs is as follows:

  CLASSIFICATION COMPARTMENT COMPARTMENT COMPARTMENT....
  CLASSIFICATION/COMPARTMENT/COMPARTMENT/COMPARTMENT....

A.1.1.2 Information Labels

- The syntax for output of information labels to the security banner on all CMWs is as follows:
  CL CW M REL C

  Where CL is classification, CW is zero or more blank-separated code words, M is zero or more blank-separated non-code-word markings, and C is either a single country code or multiple country code separated by slashes. The long form of classification, code words, and markings are used for output. The short form of classification, code words, and markings may be used otherwise (e.g., list file command). Code words, markings, and "REL" country codes are displayed in the order in which the words appear in the encodings file.

Example: TOP SECRET BRAVO1
SECRET ALPHA1 NOFORN
TOP SECRET B SB REL UK
SECRET ORCON ORG X REL UK/CAN/AUS

- The syntax for input of information labels on all CMWs is as follows:

  CL CW M REL C

  Where CL is the classification, CW is zero or more blank-separated code words, M is zero or more blank-separated non-code-word markings, and C is either a single country code or multiple country codes separated by slashes. The short or long form of classification and markings may be used for input. When entering an information label, code words, markings, and REL" country codes may be entered in any order.

  Example: TS B1
  TOP SECRET B1
  S A1 NF
  TS B1 REL UK
  TOP SECRET A SA REL UK
  S OC OX REL UK/CAN/AUS
  SECRET OC OX REL UK/CAN/AUS

A.1.1.3 Information and Sensitivity Labels Together

- The syntax for output of sensitivity labels and information labels together (as in the Classification Bar of each base window) is as follows:

  CL CW M REL C [CL COMPARTMENT COMPARTMENT...]

  Where CL is the classification, CW is zero or more blank-separated code words, M is zero or more blank-separated markings, and C is either a single country code or multiple country codes separated by slashes.

  The information labels are always followed by two or more blanks, followed by the sensitivity label enclosed in square brackets. In sensitivity labels, the short form of the classification is used. In information labels, the long form of the classification is used. The long form of the code words and/or markings is used in the information label. If the information and sensitivity labels are to be output to other than the Classification or Input Information Labels (e.g., list files with labels), then the short form of classification, code words, markings, etc. may be used.

  Example: TOP SECRET A B SA [TS A B]
  SECRET A B SA SB [TS A B C D]
  TOP SECRET A B ORCON ORG X REL CAN/UK [TS A B]

- At times when both labels can be input, the user must enter the left bracket to delimit the sensitivity label. The short or long form of the code word and/or markings may be used for input of both labels.

Example:
TS A B SA [TS A B]
TS A1 B1 [TS A B]
S A B PX LD (TS A B C D)
S A B PROJECT X LIMIDIS [TS A B C D]
TS A SA OC OX REL CAN/UK [TS A B C]
TS A B ORCON ORG X REL CAN/UK [TS A B C]

A.1.2 Guidelines for Displaying Labels

The following guidelines apply for all displaying of information labels, sensitivity labels, and clearance labels.

- Capital letters should be displayed for all classifications and words in all labels.
- Blanks should be used to separate classifications from other words in all labels, except where there are multiple words that require the same prefix or suffix, in which case the multiple words should be separated from each other with slashes.
- The long name of words should be displayed in all labels.
- The long name of classifications should be displayed in all information labels.
- The short name of classifications should be displayed in all sensitivity labels and clearances.
- The classification should be displayed first, followed by the words in the same order they appear in the encodings.
- Whenever an information label is displayed, its associated sensitivity label should also be displayed.

A.1.3 Guidelines For Changing Labels

A.1.3.1 Typing Interface

The following guidelines apply to all textual interfaces that allow users to change labels:

- When typing any label or a change to any label, the user should be able to use the following interchangeably:

  Upper and lower case letters
  Short and long names for classifications and words
  Blanks and slashes

- The following syntax should be accepted for typed changes to any label:

  [+][CLASSIFICATION] [[+]-][WORD]...

  where brackets denote optional entries, "+" or "-" denotes changes, and "..." denotes zero or more of the previous bracketed entry preceded by blanks. If the input starts with a classification followed by "+" or "-," the new classification should be used but the rest of the old label should be retained and modified as specified in the input.

- It should never be possible for the user to change the value of an information label above that of its associated sensitivity label without first, or concurrently, requesting that the sensitivity label be raised appropriately.

- The following syntax should be accepted when the user can change both the information and associated sensitivity labels of an object:

  - New information label or changes - when changes are only to information label
  - [New sensitivity label or changes] - when changes are confined to sensitivity label
  - New information label or changes [New sensitivity label or changes] - when there are changes to both information and sensitivity labels, or when user wants to set sensitivity label to same level as information label.

- The user should be shown the label resulting from the changes and asked to confirm them before they are finally made by the system.

- The existence of classifications and words for which the user is not cleared should be hidden from the user by treating such classifications or words in a manner identical to classifications or words that are not defined in the encodings.

- Whenever a user enters multiple hierarchically related words in the same label, only the highest of the words should remain in the label.

- Whenever a user enters multiple words that cannot be combined in the same label, only the first of the words specified should remain in the label.

- To the maximum extent unambiguously possible, errors made in typing changes to labels should be corrected, with error messages to the user.

A.1.3.2 GUI

The following guidelines apply to GUIs that allow label changing via the selection of individual classifications and words. Whenever reference is made in this section to a mouse, other similar pointing devices (e.g., trackballs) are also acceptable.

- The graphical interface should be integrated with the typing interface, such that the user can specify changes using either the mouse or by typing.

- A character string representation of the label should be visible after each mouse selection.
- Only classifications and words that are valid for the user to select for a label or are required in the label should be displayed in association with that label.
- Each classification or word should be annotated to indicate whether or not the classification or word is present in the label.
- Each classification or word should be separately annotated if it cannot always be selected.
- Each change specifiable through typing should have an analogy in the GUI.
- When displayed as selections in the GUI, classifications should be visually separated from words, words should appear in the order specified in the encodings, and the first pure marking word should be visually separated from the previous words. These separations should be accomplished without identification of the various components on the display (e.g., classifications, compartments, markings) because there is no universally accepted identification terminology.

If all potential selections cannot fit on the screen, a scrolling or paging mechanism should be implemented to display all selections.

A.2 WINDOW STANDARDIZATION

Window standardization is necessary to avoid confusion when users move from one CMW to another and to simplify training. To provide a standard user interface, CMW vendors will comply with the following guidance.

A.2.1 Screen Presentation

- When the machine is turned on (and before CMW is booted), user will be presented with a distinctive screen (Trusted Path) either through color, screen marking, font, or combination thereof. The user authorized to boot the CMW will be presented with a prompt to enter an appropriate user-id and, if validated, password (see Figure A-1). After booting, a user log-in screen will be displayed.
- The CMW will be configurable so at least three windows can be displayed upon user authentication.
- Icons will default to the lower left portion of the screen, starting from the lower left corner and moving right as more icons are created.

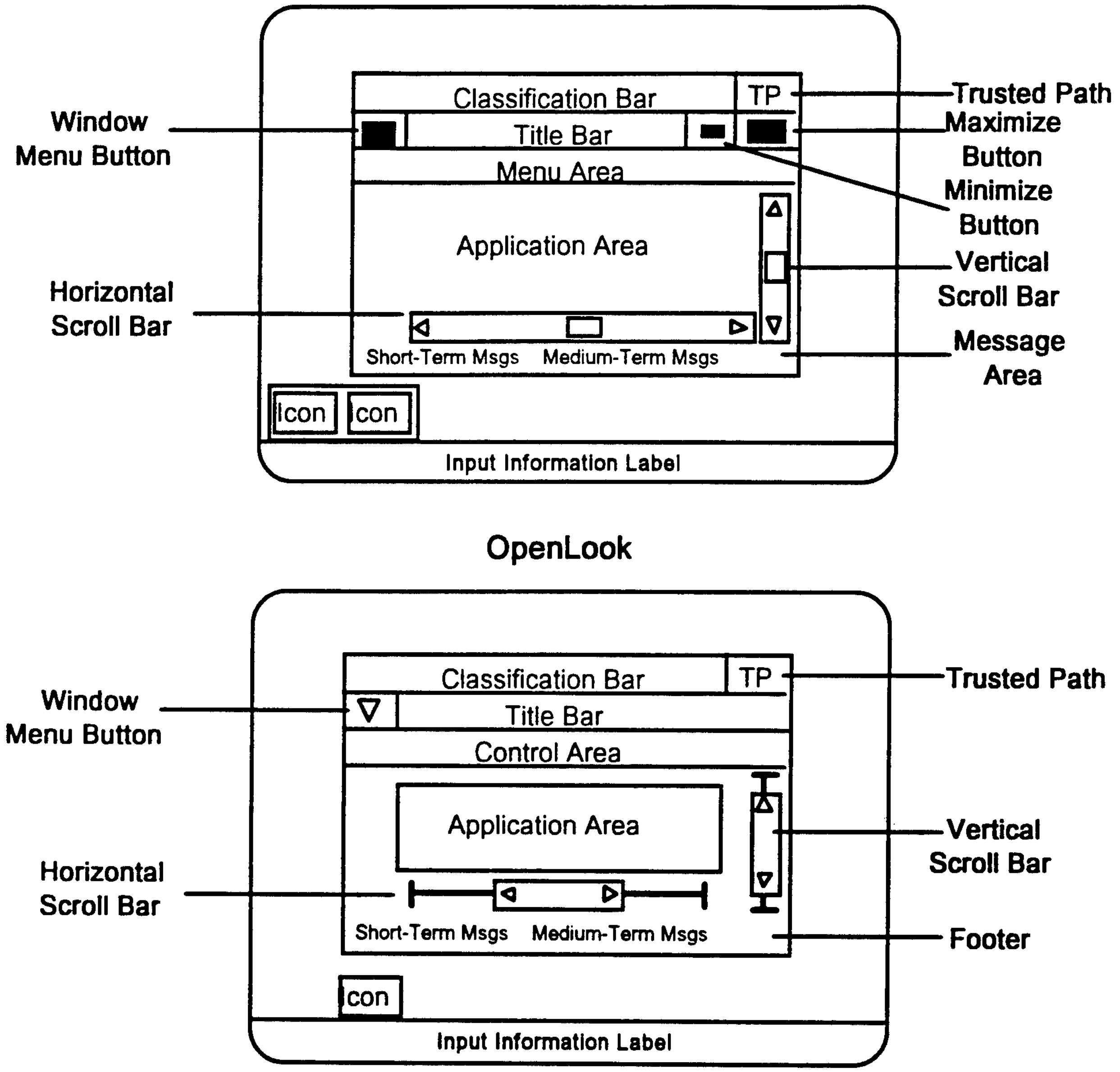


Figure A-1. Sample CMW Screens

A.2.2 Classification Bar

- A Classification Bar for output of sensitivity and information labels will be applied to each application base window created by the CMW. The Classification Bar will appear directly above the Title Bar.

- A Trusted Path button will be placed in the Classification Bar at the far right of the window. The Trusted Path button will be distinguishable by using the same mechanism (screen marking, color, font, or combination thereof) used for Trusted Path at initial log-in.

- If the information label, spacing separator, sensitivity label, and Trusted Path button are longer than the space provided in the Classification Bar, the window manager will provide the following default for trimming the labels:

  - Labels will be truncated from the right beginning with the information label and followed by the sensitivity label.

  - Truncation of the information label will be denoted by a dash followed by a greater than symbol “->“ on the left side of the label. Truncation of the information label will stop when only one character of the label remains.

  - Truncation of the sensitivity label will be denoted by a less than symbol followed by a dash “<-” on the right side of the label. Truncation of sensitivity label will stop when only one character of the label remains, not including the left square bracket ([ ).

    Example: TOP SECRET A B SA SB NOFORN [TS A B]
    ->TOP SECRET A B SA SB [TS A B]
    ->TOP SECRET [TS A B]
    ->TOP [TS A B]
    ->T [T<-

- The user must be able to view the entire information label by positioning the pointer in the classification bar and pressing any mouse button, or by menu selection via the Trusted Path.

A.2.3 Reserved Area of the Screen

- One of the CMW Trusted Path requirements is that the CMW will “provide reserved portions of the screen to which user processes cannot write.” Normally, the Input Information Label will be displayed in the reserved area. Additionally, a visual indicator (e.g., blinking or highlighted Trusted Path symbol) should be displayed in this area when the Trusted Path is active (e.g., when a window classification is being changed). The Trusted path menu can be invoked by moving the pointer to the Reserved Area and depressing the Motif Custom or Open Look Menu mouse button.

- If the Input Information Label is too long for the space provided, the window manager will provide the following default for trimming the label:

  - Input Information Labels will be truncated from the right.

  - Truncation of the Input Information Label will be denoted by a dash followed by a greater than symbol “->“ on the left side of the label.

    Example: TOP SECRET A B SA SB
    ->TOP SECRET A B SA SB PROJECT X/Y LIMDIS ORCON O

- The user must be able to view the entire Input Information Label by positioning the pointer in the reserved area and pressing the select mouse button, or by menu selection via the

Trusted Path menu. Also, upon selecting the appropriate menu item, the user may change the Input Information Label for the active window using a dialog box.

A.2.4 Trusted Path Button

- When the user clicks on the Trusted Path button, a pop-up menu with the following minimum options will be displayed in the center of the screen (see Figure A-2):

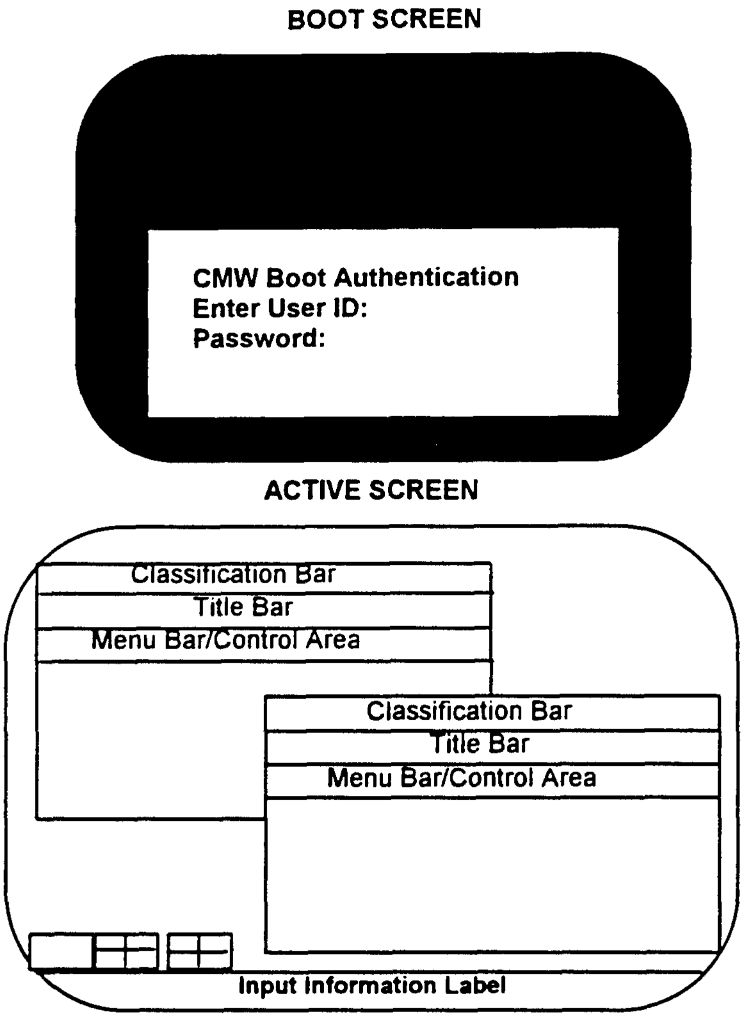


Figure A-2. Sample Trusted Path Main Menu

- Create New Window (specifying the sensitivity level and input level or letting it default to the current window)

- Change Password

- Change Information Label of a file (not of the current window)

- Change Input Information Label

- User Authorization ->.

- When a user selects "User Authorization->," a cascading menu will be displayed. The contents of the menu will depend on the individual's USER-ID (whether a "normal" user, Information System Security Officer [ISSO], Systems Administrator, or Operator), and the authorizations given that individual from the Trusted Facility Management. If the user clicks outside the bounds of the user authorization menu, the display will return to the Trusted Path main menu.

- When an authorized normal user (one whom the ISSO has included in the Access Control List of the privileged program) selects "User Authorization->," a cascading menu with the following minimum authorizations will be displayed.

  - No Classification Marking - allows a user to bypass the requirement for printing information labels on the top and bottom of each page of printed output. (It does NOT permit the user to bypass the requirement for a print banner at the beginning and end of the output.)

  - Change Sensitivity Label - allows a user to change the sensitivity label of a file the user owned.

  - Set Sensitivity = Information Label - allows a user to set the sensitivity label of a file the user owns to the information label.

- When the ISSO selects "User "Authorization->," a cascading menu with the following minimum options will be displayed:

  - .. Change Sensitivity Label - allows the ISSO to set the sensitivity label of any file (file, device, file system, etc.) on the CMW.

  - Assign User Password/Clearance - allows the ISSO to set the password, password length, password time-limit, and clearance of a user's profile.

  - File Privileges - allows the ISSO to set/show the privileges associated with files.

  - Operator Administrator Authorization - allows the ISSO to set/show the authorizations for the other users, the administrator, and operators.

  - Set Audit Events - allows the ISSO to configure the auditing system.

  - Review Audit Data - allows the ISSO to reduce and review the audit data and store the audit data on removable media.

- When the System Administrator selects "User Authorization->," a cascading menu with the following minimum options will be displayed:

- Add User - allows the administrator to create a user profile, assign the user to a group, and create a user home directory. The ISSO must then assign the user a password.
- Add Group - allows the administrator to create a discretionary access group.
- ISSO Authorization - allows the administrator to set the authorizations of the ISSO.

- When the operator selects "User Authorization->," a cascading menu with the following minimum options will be displayed:
  - Make File System Backups - allows the operator to back up file system on the CMW.
  - Configure Printer - allows the operator to configure the printer (maximum sensitivity, printer definition, communication port, etc.).
  - Enable Printer - allows the operator to enable printer service to users.
  - Disable Printer - allows the operator to disable a printer for reconfiguration, etc.
  - Mount a file System - allows the operator to mount a file system on the CMW.
  - Unmount a File System - allows the operator to unmount a file system from the CMW.
  - Halt System - allows the operator to halt the CMW.

A.3 CLASSIFICATION DISPLAY ENHANCEMENTS

This guide designates a minimum security functionality, which the vendor is encouraged to supplement. An example could be to provide a menu box from the Trusted Path that builds an information label for a specified file (see Figure A-3). Clicking a mouse button over the classification, multiple code words, multiple handling caveats, and multiple release markings could dynamically build the appropriate information label.

Another example could be to implement Trusted Path differently depending on whether it was invoked using the Trusted Path button or through the Trusted Path Background. The menu from the Trusted Path button could present options related to windows (Create, Cut/Paste, etc.) and the menu from the background selection could present non-window related items (Change Password, User Authorizations, etc.).

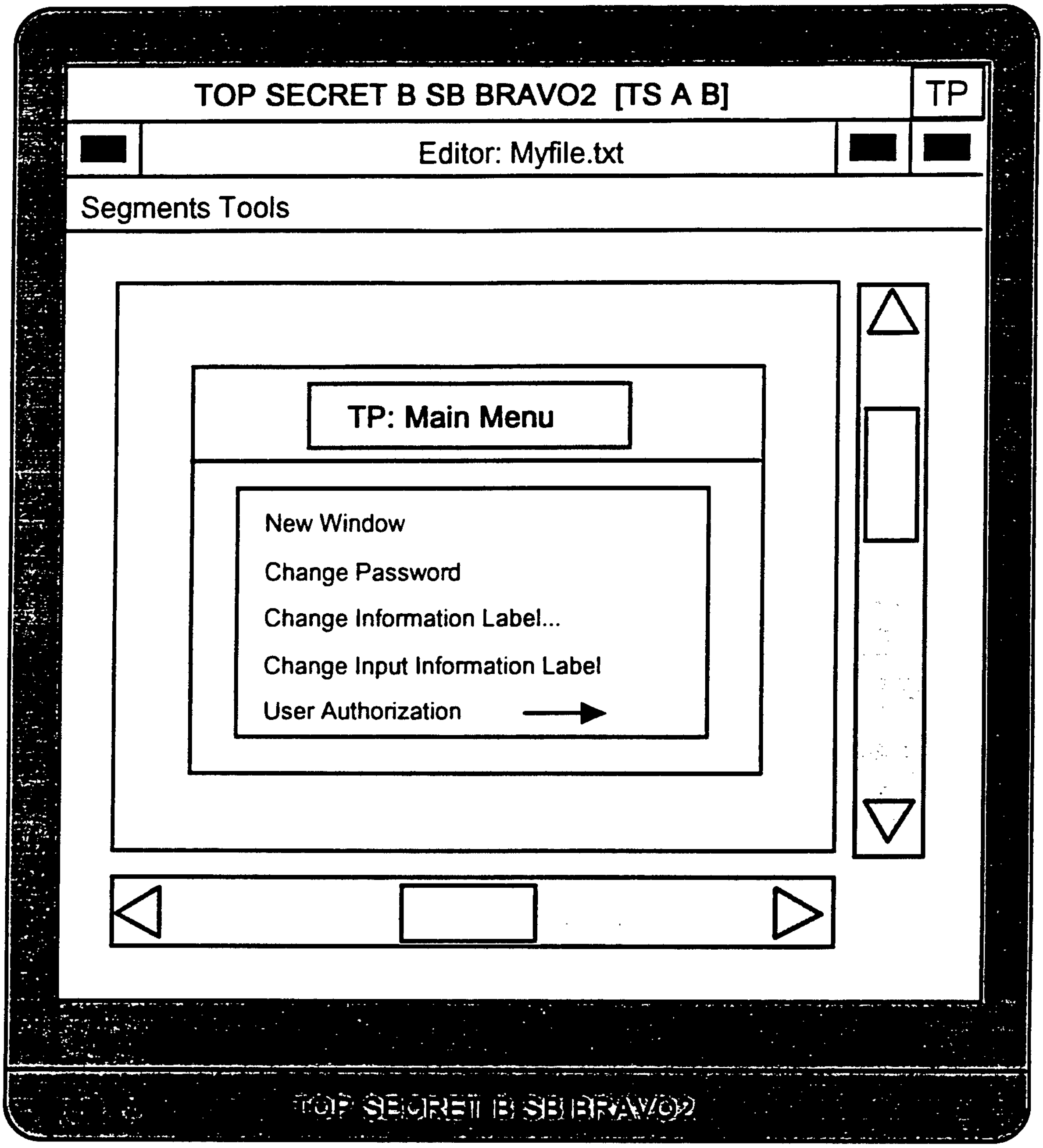


Figure A-3. Sample for Entering New Input Information Label

This page intentionally left blank.

APPENDIX B

GLOSSARY

This page intentionally left blank.

APPENDIX B

GLOSSARY

AC: See alternating current.

accelerator key: Special key or key combination that performs the same action as a menu selection

Ada: High-level computer programming language developed by the Department of Defense (DoD). Ada is used as the standard programming language for DoD. It is used for real-time processing, is modular in nature, and includes object-oriented features.

alternating current (AC): Electrical current that reverses its direction at regularly recurring intervals

API: Application Programming Interface

APP: Application Portability Profile

application: Classification of computer programs designed to perform specific tasks, such as word processing, database management, or graphics

applications menu: List of options within an application

automated tools: Software performing a sequence of operations to assist the user in achieving a goal (e.g., within graphics software, functions that align objects, smooth curves, or draw circles)

backlighting: Lit from the back. When referring to a monitor, light passes through the display screen from the back in order to illuminate screen images.

base-level functions: Initial or basic functions

batch processing: Processing data or the accomplishment of jobs accumulated in advance in such a manner that each accumulation thus formed is processed or accomplished in the same computer run

baud: Measure of the transmission speed capability of a communications line or system. In a sequence of binary signals, the rate of one baud equals one bit per second.

bit-mapped display: Display in which every picture element (pixel) of the screen can be referenced individually

bookmarking: Method of tagging items of interest to the user for easy referral later. Allows the user to customize the application.

Boolean logic: Logical expression that uses Boolean operators such as AND, OR, NOT, XOR, NOR, and NAND to create a statement that, when resolved, is either true or false

Boolean operators: A keyword in programming that causes two values to be combined in a logical fashion

branching menu: Menu that, if selected, brings up another menu

bring-to-front: Process of moving a window to the foreground

Candela (cd): Unit of luminous intensity expressed in Candela per square meter (cd/m^2). One cd is equal to 0.29 footLambert.

CAP: Computer/Electronic Accommodation Program

Cathode ray tube: Electronic vacuum tube that focuses electrons energizing phosphors on a screen, creating a visible display. The typical computer monitor uses this type of display technology.

cd: See candela.

central processor: Portion of the computer that controls execution of applications

character: Single letter, digit, or symbol

character string: Series of alphanumeric characters, the contents of which are treated as though they were text

CIE: Commission International d'Eclairage

CMW: Compartmented Mode Workstation

COBOL: Acronym for Common Business-Oriented Language. COBOL is a computer programming language used extensively in mainframes and minicomputers for business applications.

command: Entry that instructs the computer to effect a specific action

command entry: Informing the computer that a specific command should be effected

command icons: Computer icons that represent frequently used computer commands and operations

command language: Limited programming language used strictly for executing a series of commands

command stacking: Allows the user to key a sequence of commands as a single "stacked" command entry

compatible letters: Letters easily associated with the function requested, for example, "P" for print, "Q" for quit

context-sensitive: Computer action or response directly related to the cursor position or specific point in the software, for example, a help function that displays information about the specific data entry field in which the cursor was located when help was called.

control action: Actions that must be effected to control a window or other graphics object or its contents

control entry: Input action by the computer user that changes some aspect of the appearance or function of the application

control lockout: Processing delay that results in pacing the capability to enter sequences of control commands

COTS: Commercial Off-The-Shelf (software)

courseware: Another name for educational or training materials and software

CPU: Central Processing Unit

crosstalk: Optical crosstalk, or bleeding, occurs when the light from the incorrect video image gets through. When referring to stereoscopic images, the right eye's image is visible to the left eye or vice versa.

CRT: See Cathode Ray Tube

CWS: Compartmentalized workstation

cursor: Visual mechanism to mark, on-screen, where current input or output is to happen

data entry: Series of keystrokes used to input information into the computer

data entry field: Space (number of characters and/or digits) allowed for data entry

data field: Location in a file or database that contains a specific type of information

database: Structured or organized collection of information, which may be accessed by the computer

database management system: Computer application program that accesses or manipulates the database

DBMS: Database Management System

DC: See direct current.

default: Command that is automatically executed if none is specifically indicated

default value: Value of a variable in lieu of a specifically indicated value

defeated: Option that cannot be selected due to another selected option's use

delimiter: Symbols such as commas, spaces, or parentheses, which mark the boundaries of a specific block of information

designate: Process of selecting and displaying the current or active window with visual cues

destructive entries: Any entry that will destroy or overwrite information

DIA: Defense Intelligence Agency

dialog: Structured series of interchanges between a user and a computer terminal. Dialogs can be initiated by the computer or the user. Interactive dialog consists of an action by the user followed by a response from the computer or vice versa.

dialog box: Screen display box containing a message requesting additional information from the user

direct current (DC): Electrical current that flows in one direction only and is substantially constant in value

direct manipulation: Method of data organization (typically involving extensive windowing and iconization) in which the user can select specific displays of information and move them about to facilitate interaction with an application. A system of interaction in which the user's actions directly affect software operations.

DISA: Defense Information Systems Agency

display frame: Window or page

display parallax: When used in discussing touch screen technology, display parallax is the apparent displacement of an object viewed on a curved CRT screen and seen through a flat touch interactive display.

display screen: Screen of a multipage file

DMA: Defense Mapping Agency

DoD: U.S. Department of Defense

DODIIS: Department of Defense Intelligence Information Systems

double keying: Each character of the data item does not have an appropriately labeled key and therefore requires more than one keystroke for entry.

DTED: Digital Terrain and Elevation Data

dual activation: Two key are used simultaneously to input a command.

dynamic depth displays: Stereoscopic displays that are designed to change (move) images during viewing

electroluminescence (EL): Luminescence produced by electrical excitation of phosphor in powder or film form

electronic mail: Communication, processed through a network, from one workstation to another

end user: Person who ultimately uses the computer application or output

error management: Various options within an application that allow the user to eliminate the effects of commands executed accidentally or unwisely

expand: Ability to resize objects to produce better organization of on-screen material, usually a graphic or a window

fc: See footcandle.

feedback: Visual acknowledgment that the computer is executing the command or that the command was executed

field: Addressable data location

file: Any specifically identified collection of information stored in the computer

FIP: Federal Information Processing

FIPS: Federal Information Processing Standard

FIRMR: Federal Information Resources Management Regulation

Fl: See footLambert.

footcandle (fc): Unit of measurement of illumination. The amount of light emitted by a standard candle (1 cd) measured one foot away from the candle equals one footcandle.

footLambert (Fl): Unit of measure of intensity of reflected or emitted light (luminance). The average luminescence of any reflecting surface in footLamberts is the product of the illumination in footcandles by the luminous reflectance of the surface.

frame: Single display image or screen

function key, fixed and variable: Key which, when depressed, effects a specific action. It can either be a single, predefined function (fixed), or vary according to the system mode or level within an interactive dialog.

form filling: Method of interaction in which the user enters a series of commands or data items in predefined fields. These fields may be mandatory or optional.

FORTRAN: Acronym for FORmula TRANslator, which is a high level computer language used extensively in scientific and engineering applications

freeze: See Option - PAUSE

GENSER: general security

GIS: Geographic Information Systems

GOTS: Government Off-The-Shelf (software)

graphical interaction: Transactions between the user and computer-generated graphical representations of objects (screens, menus, buttons, etc.)

Graphical User Interface (GUI): System design that allows the user to effect commands, enter into transaction sequences, and receive displayed information through graphical representations of objects (menus, screens, buttons, etc.)

GUI: See Graphical User Interface.

hard copy: Printed copy of machine output in a visibly readable form, for example, printed reports, listings, documents, summaries

hardware architecture: Assemblage of a computer's internal components and its attached peripheral devices, which determine its capabilities and its limitations

hatching: Graphical pattern characterized by 45 and 135 degree diagonal lines that cross the patterned area

HCI: See Human Computer Interface.

help screen: Separate window that offers advice and information on how to overcome a specific problem and/or to better interact with the computer

HFE: See Human Factors Engineering.

hierarchical menu: Method of organizing menus in layers. The secondary or tertiary menus are stored within a primary menu.

high level language: Programming language that does not reflect the structure of any one computer or class of computers

high resolution: Screen display within an extremely fine visual reproduction of detail

highlight: Visual method to call attention to a specific piece of text or a graphic through differentiating it from surrounding texts or graphics. This is usually accomplished using contrasting colors or reverse video.

hook: Selecting a corner of a window or icon in order to move or resize it

Human-Computer Interface (HCI): Hardware and software allowing information exchange between the user and the computer

Human Factors Engineering (HFE): Approach that makes use of scientific facts in the design of items (i.e., computer systems, software, etc.) to produce effective human-machine integration and utilization

icon: Graphical representation of an object, concept, or message used by a computer system to represent items such as files, documents, programs, and disk drives.

iconify: Process that changes the text representation of an object, concept, or message into an icon

iconification: Process of iconifying

IEEE: Institute of Electrical and Electronics Engineers

illuminance: Measure of the quantity (density) of light reaching an object or surface. Measured in footcandles.

Image Formation Time (IFT): Measurement of the time required to update screen image displays

Infrared (IR): Radiation outside the visible light range on the red side (wavelength 0.75 to 0.8 micrometers)

input focus: Applies to a window that actually receives user input. This window is known as the active window where keyboard input appears and pointing device inputs apply. "Explicit" input focus refers to user or application action (e.g., typing keyboard accelerators, clicking pointer inside a window, moving a window through menu selection, etc.) to assign input focus. "Implicit" focus refers to focus automatically assigned to the window containing the location cursor.

interactive control: Attribute describing the ability of a program and a user to interface with each other during program execution

interactive dialogue: See dialog.

interactive procedures: Methods by which a user interacts with a computer and the computer with the user

interface: Interconnection and interrelationships between two devices, two applications, or the user and an application or device

interlock: Mechanism to connect two or more processes within a computing system to ensure that no one part of a hardware or software system can be operated independently

interocular: Perceptual coordination between the eyes

IFT: See image formation time.

IR: See infrared.

JMCIS: Joint Maritime Command Information System

JPEG: Joint Photographic Experts Group

jump-ahead: Capability of moving ahead during a step-wise process to allow quicker performance of an operation; useful for experienced computer users.

justification: Alignment of text on a display or a printed page. Left justification means that the left margin is even.

keyword: Special word in a programming language that tells the computer which operation to perform

Lambert: See footLambert.

landscape: Screen display or printing orientation parallel to the wide side of the paper

LCD: See Liquid Crystal Display.

LCSS: See Liquid Crystal Stereoscopic Shutter.

left-justified: See Justification.

Liquid Crystal Display (LCD): Display operated by polarizing light in which the nonactive segment reflects incident light and thus appears invisible against its background

Liquid Crystal Stereoscopic Shutter (LCSS): Type of display that utilizes liquid crystal shutters, one for each eye synchronized to alternate fields of the display, and representing one of the two images necessary to achieve the third dimension

lockout: Condition of the application locking the keyboard (i.e., not accepting commands from it) while the application is executing a command

log on: Process of gaining access to the system, usually involving a password and a recognition of the specific user by the computer

logarithm: The exponent that indicates the power to which a number has been raised to produce the given number:

$$N = 10^n \qquad \log_{10} N = n$$

luminance: Amount of light per unit area reflected from or emitted by a surface. Measured in footcandles.

lux: Standard measure of illuminance. One lux is one lumen per square meter.

macro: Executable file that stores a series of commands and keystrokes to be used later

MANpower and PeRsonnel INTegration (MANPRINT): An Army program that addresses concerns with manpower, personnel, training, human factors, system safety, and health hazards

MANPRINT: Acronym; MANpower and PeRsonnel INTegration

masking: Partial or complete obscuring of one item by another

memory: Place in the computer in which information is stored

menu: List of options available within a software application

menu bar: The horizontal menu, usually at the top of the screen, which contains menu titles

metaphor: System-level analogy used for the grouping of processes and/or procedures. Usually associated with icons based on the analogy. As, for example, a desk top metaphor where icons represent office equipment or operations.

minimize: Procedure to make the window as small as it can be without being closed; this is usually done through iconization.

mnemonic: Word or code symbolic of another word, code, or function

mode: Status of the screen or program process

Modulation Transfer Function (MTF): A parameter using spatial frequency responses to characterize a screen display. The spatial frequency is stated in lines (line pairs) or minimum/maximum intensity pairs per unit distance. The MTF is used as a performance measurement of many optical systems.

Motif: User interface design approach based upon the "look" and "feel" presented in the OSF/Motif™ style guide. Motif™ is marketed by the Open Software Foundation.

MTF: See Modulation Transfer Function.

multifunction keying: Interface design where computer keys may perform multiple functions with the use of a combination of keystrokes

multiwindow: Simultaneous display of several windows on the computer screen

natural language: Programming language paradigm exemplified by using English-like commands and syntax to issue commands; interactions in the vernacular of the user.

navigation: Manner in which the user moves through the menu structure

NATO Forces: Personnel in the military forces of member nations of the North Atlantic Treaty Organization (NATO)

NIST: National Institute of Standards and Technology

nit: See normalized intensity.

normalized intensity (nit): Metric unit of measure of luminous intensity. A nit is equal to one candela per square meter (cd/m^2) or 0.29 footLambert.

null: Empty; nothing. A null set contains no elements.

OCR: See Optical Character Recognition.

one to many mapping: An icon that represents a category of possibilities within an option is a one to many mapping.

one to one mapping: An icon that represents a single, specific function is a one to one mapping.

OOP: Object Oriented Programming

Open Systems Environment: See OSE.

Open Software Foundation (OSF): Consortium of computer hardware and software manufacturers whose membership includes over seventy of the computer industry's leading companies

open window map: A map (graphic display) that shows windows that are open and how they relate to each other

open: Procedure to cause a window to be displayed from an icon or menu option so that a document, directory or file can be viewed

Optical Character Recognition (OCR): The analysis and translation of a graphic

representation of text into a coded form, such as ASCII or EBCDIC

option: Command that may be selected to access a specific function of an application

option - BACKUP: Option that will display the last transaction or the process of saving information to non-volatile memory

option - CANCEL: Command that allows the user to have the computer disregard a previous command

option - CONFIRM: Explicit warning of any possible data loss

option - CONTINUE: Option that resumes a transaction sequence which has been stopped by a PAUSE

option - GOBACK: Option that will display the last transaction. See also BACKUP.

option - PAUSE: Option that temporarily causes a transaction sequence to stop running. Use the CONTINUE option to resume after pausing.

option - RESTART: Option that will cancel any entries that have been made in a transaction sequence and return to the beginning of a sequence

option - REVIEW: Option that returns to the first display of a transaction sequence, allowing the user to review the transaction and make necessary changes

option - SUSPEND: Option that allows a user to leave the application, then, when he/she returns, resume at the same point he/she left off

option - UNDO: Option that immediately reverses any action

option code: Codes associated with the available choices

OS: Operating System

OSE: Open Systems Environment

OSF: See Open Software Foundation.

output: Information the computer displays in response to the user's actions

overarching guidelines: Dominant or all-embracing guidelines

overlapping: Windowing system in which one window covers a portion of another

overlay: Printing or drawing on a transparent or semi-transparent medium on the same scale as a map, chart, etc., to show details not appearing or requiring special emphasis on the original

paging: Scrolling through material one page at a time

paired opposites: Set of opposite functions, such as up and down, top and bottom

pan: Process to change the displayed region (often of a map) in a regular and smooth manner

parallax: Apparent displacement of an object as seen from two different points not on a straight line with the object

parameter: Quantity or constant whose value varies with the circumstances of the application

piezoelectric: Electric polarity due to pressure, especially in a crystalline substance

pixel: Contraction for picture element. A pixel is a single dot on a display screen.

pixel matrix: Arrangement of screen dots (pixels) to form text or graphic displays

pop-up menu: Lists of options that appear on the display screen in the form of a window

portrait: Screen or printing orientation parallel to the narrow side of the paper

predictive modeling: Use of a model to predict the actual response of a system or process

preformatted: Screen structure prepared for the user

presentation graphics: Pictorial representations of the relationships between variables (graphs and charts) or representations of systems (diagrams, schematics, and graphical renditions)

primitive: Code that defines a specific elementary shape, form, or color

programming language: Artificial language established for expressing computer programs

prompt: Text or graphic display that indicates the start point for user-generated actions. This term is also used for software generated instructions for process confirmation.

pull-down menu: Lists of options attached to a selection on a menu bar

push-to-back: Process of moving a window to the background

QBE: Query by Example

QBF: Query by Forms

query language: Specialized type of command language to elicit information from the computer system

real time: Absence of delay, except for the time required for transmission

real-time control system: Systems capable of responding to external events with negligible delays

resize: Procedure to change the size of a window or graphic

resize border: Window border that, if selected, allows user to resize the window

resolution: Density of picture display elements of the screen; degree of detail with which an image is displayed or printed.

retrieve: Procedure required to display stored information for purposes of viewing and manipulation

RGB: The original color display for the IBM PC (the Personal Computer Color Display IBM model 5151) used three discrete digital signals for each of the three primary colors. From these signals, the display type earned the nickname RGB from the list of additive primary colors: Red, Green, and Blue. Except for the interface signal, the RGB monitor works like a composite color monitor, using the same frequencies but substituting digital signals for analog.

right-justified: See justification.

SAW: See Surface Acoustic Wave.

scroll: Method used to move the contents of a window or list in a dialogue box using the scroll bar or scroll arrows

scroll bar: Rectangular bar that may be along the right edge or bottom of a window. Clicking or dragging in the scroll bar causes the view of the document to change.

secondary coding: Providing more than one method for coding displayed information. For example, in coding a particular item with red color, the use of the symbol "R" would provide secondary method for conveying the information when color was not available.

semantics: Relationship of characters or groups of characters to their meanings, independent of the manner of their interpretation and use

sensitivity analysis: Study that shows the response of a system to varying conditions. For example, "How sensitive is the system to increased workload?"

sequence control: Prescribed control over the order of function performed by the computer; this impacts the way in which a user interacts with the application.

size coding: Variations in the size of displayed alphanumerics and symbols. Such coding can be used for categorization.

slider: Part of the scroll bar that indicates what part of the file contained in a window is being viewed

soft keys: Visual representation of key functions on the display screen. This is usually associated with software controlled function key capabilities.

specular reflector: Reflecting light in a diffuse manner

SQL: Structured Query Language

stacked command: Single command composed of multiple commands that must be executed individually

stereopsis: Phenomenon of simultaneous vision with two eyes in which there is a vivid perception of distance of objects from the viewer (three-dimensional or stereoscopic vision)

stereoscopic: Method of seeing objects in three dimensions

stroke width (sw): Width of the line used to create a displayed character

subordinate window: A window that is opened from and controlled by another window

subtend: Opposite in position

summary symbols: Symbol that categorizes the information portrayed by a group of symbols

supraordinate window: Higher level window, usually the window from which subordinated options or tasks are controlled

Surface Acoustic Wave (SAW): When used in the context of touch screen technology, an approach that uses ultrasonic sound "beams" transmitted from two perpendicular sides of a display frame.

sw: See stroke width.

system level menu: List of which applications are available for utilization

system response time: Amount of time that elapses between a command being given and its being executed by the computer

text editor: Application that allows text to be created or modified

text-based systems: Method of organization in which the primary form of interaction between the system and user is through text rather than through graphical or voice interaction

three-dimensional: Relating to the three physical dimensions (height, width, depth). Giving the effect of depth or varying distances.

TID: See Touch Interactive Display.

tiling: Windowing approach in which multiple windows do not overlap, rather, all lie on the same plane.

theater-type displays: Display screens suitable for large group presentations as used in a movie theater or auditorium

title banner: Horizontal bar at the top of a window that shows the name of the window and allows it to be moved

Touch Interactive Display (TID): Uses a physical device between the user and the display which acts as the input mechanism

transaction: Interaction between a user and a computer in which the user inputs a command to receive a specific result from the computer

transaction sequence: Order of transactions required to accomplish the desired results

transmissivity: Measurement of the ability of an image to be transmitted. When used in the context of touch screen technology, refers to the ability of the image to be transmitted through a filter placed in front of a computer screen.

type-ahead: Capability of the computer to receive commands faster than it can display their results

UAPI: Uniform Application Program Interface

UCI: See User-Computer Interface

UIDL: User Interface Definition Language

UIMS: User Interface Management System

user-callable: Able to be requested by the user as desired

User-Computer Interface (UCI): Hardware and software allowing information exchange between the user and the computer

user-specified windows: Windows whose content has been selected by the user

variable: Quantity that can assume any of a given set of values

VDT: See Video Display Terminal.

Video Display Terminal (VDT): Terminal composed of a keyboard for data input and a CRT screen for display of the input/output

widget: Basic graphical object, which is a component of a user interface component

window: Typically rectangular display that provides a visual means for interaction with an application

zoom: Graphical tool used to magnify a portion of a document for more detailed viewing

This page intentionally left blank.

APPENDIX C

REFERENCES

Note: References appearing in this section represent documents used in preparation of this volume, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

41 CFR. *Federal Information Resources Management Regulations (FIRMR),* Chapter 201. Government Printing Office (GPO), Washington, D.C.

Abramson, S. R., L. H. Mason, and H. L. Snyder. 1983. "The Effects of Display Errors and Font Styles Upon Operator Performance with a Plasma Panel." In *Proceedings of the Human Factors Society - 27th Annual Meeting*, pp. 28-32. HFS, Santa Monica, California.

ACM - see Association for Computing Machinery, Inc.

Air Force Intelligence Agency. 1990. *Air Force Intelligence Data Handling System Style Guide.* Air Force Intelligence Agency, Washington, D.C.

Ambron, S., and K. Hooper. 1990. *Learning with Interactive Multimedia: Developing and Using Multimedia Tools in Education.* Microsoft Press, Redmond, Washington.

American Institute of Graphic Arts (AIGA). 1982. *Symbol Signs.* Visual Communications Books, Hastings House, New York.

Andrews, J. R., W. E. Haas, and M. D. Rainsdon. 1989. "Holographic Displays for the Man-Machine Interface." *Optical Engineering*, 28(6):643-649.

Andriole, S. J., and L. A. Adelman. 1990. "Prospects for Cognitive Systems Engineering." Presentation, Center for Multidisciplinary Information Systems Engineering, Drexel University, Philadelphia.

Antin, J. 1988. "An Empirical Comparison of Menu Selection, Command Entry and Combined Modes of Computer Control." *Behavior and Information Technology* 7(2) 173:173-182.

Apple Computer, Inc. 1992. *Macintosh Human Interface Guidelines.* Addison-Wesley Publishing Company, Reading, Massachusetts.

Arend, U., K. Muthig, and J. Wandmacher. 1987. "Evidence for Global Feature Superiority in Menu Selection by Icons." *Behaviour and Information Technology* 6(4):411-426.

Armstrong, H. G. 1988. "Colored Light-Emitting Diodes: Use of Red or Green in High Ambient Illumination." *In Engineering Data Compendium: Human Perception and Performance,* eds. K. R. Boff and J. E. Lincoln, Vol. 3:2260-2261. Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Arnstein, J. 1983. *The International Dictionary of Graphic Symbols.* Whitstable Litho Ltd., Whitstable, Kent, England.

Aspillaga, M. 1991. "Screen Design: Location of Information and Its Effects on Learning." *Journal of Computer-Based Instruction* 18(3):89-92.

Association for Computing Machinery, Inc. (ACM). 1990. *Resources in Human-Computer Interaction.* ACM Press, New York.

Association for Computing Machinery, Inc. (ACM). 1989. *Human Factors in Computing Systems: CHI '89 Conference Proceedings*, Austin, Texas. ACM Press, New York.

Auerbach Publishers, Inc., ed. 1981. *Practical Data Base Management.* Reston Publishing Company, Inc., Reston, Virginia.

Avery, L. W., R. V. Badalamente, S. E. Bowser, P. A. O'Mara, and S. E. Reynolds. 1990. *Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface, Version 1.0.* Pacific Northwest Laboratory (PNL) for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.

Avery, L. W., S. E. Bowser, S. M. Adams, R. V. Badalamente, D. T. Donohoo, D. A. Gellert, K. D. Hargrove, W. A. Hesser, J. G. Heubach, P. A. O'Mara, D. J. Pond, R. B. Randall, S. E. Reynolds, and M. S. Rowley. 1992. *Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface, Version 2.0,* eds, L. W. Avery and S. E. Bowser. Pacific Northwest Laboratory (PNL) for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.

Aykin, N. M. and T. Aykin. 1991. "Individual Differences in Human-Computer Interaction." *Computers & Industrial Engineering* 20(3):373-379.

Badler, N., and B. Webber. 1991. "Task Communication Through Natural Language and Graphics." *AI Magazine* 11(5):71-73.

Baeker, R. 1980. "Towards an Effective Characterization of Graphical Interaction." *IFIP Workshop on Methodology of Interaction,* pp. 127-147.

Baggen, E. A., H. L. Snyder, and M. R. Miller. 1988. "A Human Factors Evaluation of Current Touch-Entry Technologies." *In SID International Symposium Digest of Technical Papers*, pp. 259-262. SID, Playa Del Rey, Kingsbeach, California.

Bailey, M. J. 1993. "Guidelines For The Use Of Color In Scientific Visualization." SIGGRAPH Course Notes. San Diego Supercomputer Center.

Bailey, R. W. 1989. *Human Performance Engineering: Using Human Factors/Ergonomics to Achieve Computer System Usability.* Prentice Hall, Englewood Cliffs, New Jersey.

Bailey, R. W. 1982. *Human Performance Engineering: A Guide for System Designers.* Prentice Hall, Englewood Cliffs, New Jersey.

Baker, C. 1988. *Text-Editing Performance as a Function of Screen Size: A Pilot Study.* 612716.H70007000l1, U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Barnett, B. J. 1990. "Aiding Type and Format Compatibility for Decision Aid Interface Design." In *Proceedings of the Human Factors Society 34th Annual Meeting*, pp. 1552-1556. HFS, Santa Monica, California.

Barten, P. G. J. 1991. "Resolution of Liquid-Crystal Displays." *In Society for Information Display 91 Digest*, pp. 772-775. SID, New York.

Beaton, R. J. 1990. "Human Factors Engineering of 3D Stereoscopic Display Systems." In *Stereographics*, 17th International Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, Special Interest Group on Graphics, Dallas, Texas.

Beaton, R. J., and S. T. Knox. 1987. "Flat Panel Image Quality." In *Society for Information Display 8⁻ Digest*, pp. 115-118. SID, New York.

Benbasat, I., and Y. Wand. 1984. "A Structured Approach to Designing Human-Computer Dialogues." *International Journal of Man-Machine Studies* 21(2):105-126.

Benson, B. L., and J. E. Farrell. 1988. "The Effect of Character Height-to-Width Ratio on CRT Display Legibility." In *Society for Information Display 88 Digest*, pp. 340-343. SID, New York.

Biberman, L. M., and B. Tsou. 1991. *Image Display Technology and Problems with Emphasis on Airborne Systems.* Technical Report No. AD-B1157 161, Defense Technical Information Center, Alexandria, Virginia.

Bidgoli, H. 1989. "DSS Products Evaluation: An Integrated Framework." *Journal of Systems Management* 40(11):27-28

Billingsley, P. A. 1988. "Taking Panes: Issues in the Design of Windowing Systems." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 413-434. Elsevier Science Publishers B.V., Amsterdam.

Blaha, R. J. 1992. *Emerging Large Screen Display Technology.* Technical Report No. M92B0000010, The MITRE Corp., Bedford, Massachusetts.

Blaha, R. J. 1990. *Large-Screen Display Technology Assessment for Military Applications.* Technical Report No. M90-16, The MITRE Corp., Bedford, Massachusetts.

Blaha, R. J., C. D. Crotty, and D. F. Martin. 1992. *Evaluation of Large-Screen Display Systems in Workstation Development Environment.* Working Paper No. 92B0000081, The MITRE Corp., Bedford, Massachusetts.

Blankenberger, S., and K. Hahn. 1991. "Effects of Icon Design on Human-Computer Interaction." *Int. J. Man-Machine Studies* 35:363-377.

Blaser, A., and M. Zoeppriz, eds. 1983. *Enduser Systems and Their Human Factors.* Springer-Verlag, Berlin.

Blattner, M. M., and R. B. Dannenberg. 1992. *Multimedia Interface Design*, ACM Press, New York.

Blattner, M. M., D. A. Sumikawa, and R. M. Greenberg. 1989. "Earcons and Icons: Their Structure and Common Design Principles." *Human-Computer Interaction* 4:11-44.

Bobko, D. J., P. Bobko, and M. A. Davis. 1986. "Effect of Visual Display Scale on Duration Estimates." *Human Factors Society*, 28(2):153-158. HFS, Santa Monica, California.

Boehm-Davis, D. A., R. W. Holt, M. Koll, G. Yastrop, and R. Peters. 1989. "Effects of Different Data Base Formats on Information Retrieval." *Human Factors* 31(5):579-592.

Bosman, D. 1991. "Simulation of the Perception of a Computer-Generated Display Image." *Proceedings of Society for Information Display*, 32(1):3-12. SID, New York.

Bowser, S. E. 1991. "Review of Army Tactical Command and Control System Soldier-Machine Interface Functional Issues." Pacific Northwest Laboratory (PNL) Task Order 91-04, Subtask 2b for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.

Breen, P. T. 1989. *Functional Requirements for C3I Large Screen Displays.* Technical Report No. M89-01, The MITRE Corp., Bedford, Massachusetts.

Breen, P. T. 1987. "Functional Requirements for C3I Large Screen Displays." In *Large Screen Projection Displays*, 760:2-8. SPIE Press, Bellingham, Washington.

Breen, P. T., and H. C. Masterman. 1990. *Modulation Depth Metrics for the Large Screen Displays in NASA's Mission Control Centers.* Working Paper No. 28736, The MITRE Corp., Bedford, Massachusetts.

Breen, P. T., P. E. Miller-Jacobs, and H. H. Miller-Jacobs. 1987. "Color Displays Applied to Command, Control, and Communications (C3) Systems." *In Color and the Computer*, ed. J. H. Durrett, pp.171-187. Academic Press, Inc., Boston.

Brockmann, R. J. 1990. *Writing Better Computer User Documentation - From Paper to Hypertext Version, 2.0.* John Wiley & Sons, Inc. New York.

Brooks, J. D., R. D. Gilson, and H. R. Myler. 1990. "Display Design Guide for Bivisual Media." *Proceedings of the Human Factors Society 34th Annual Meeting 1990.* HFS, Santa Monica, California.

Brosda, V., and G. Vossen. 1988. "Update and Retrieval in a Relational Database Through a Universal Schema." *ACM Transactions on Database Systems* 13(4):449-485.

Brosley, M., and B. Shneiderman. 1978. "Two Experimental Comparisons of Relational and Hierarchical Database Models." *International Journal of Man-Machine Studies* 10:625-637.

Brown, C. M. 1989. *Human-Computer Interface Design Guidelines.* Ablex Publishing Corporation, Norwood, New Jersey.

Brown, C. M., D. B. Brown, H. V. Burkleo, J. E. Mangelsdorf, R. A. Olsen, and R. D. Perkins. 1983. *Human Factors Engineering Standard for Information Processing Systems.* LMSC-D-877141 (Revision of C410), Lockheed Missiles and Space Company, Inc., Los Angeles.

Bullinger, H. J., K. P. Fähnrich, and J. Ziegler. 1987. "Software-Ergonomics: History, State-of-the-Art, and Important Trends." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, ed. G. Salvendy, pp. 307-316. Elsevier Science Publishers B.V., Amsterdam.

Bunnell, D., ed. 1993. *1993 Multimedia Tool Guide.* Newmedia Technologies for Desktop Computer Users, Special Issue: March 1993.

Burger, J. 1993. *The Desktop Multimedia Bible.* Addison-Wesley Publishing Company, Reading, Massachusetts.

Campbell, J. A., and S. P. Ross. 1987. "Issues in Computer-Assisted Interpretation of Graphs and Quantitative Information." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, ed. G. Salvendy, pp. 473-480. Elsevier Science Publishers B.V., Amsterdam.

Carbone, R. M., and D. MacIver. 1987. "A Survey of Large-Screen Displays of C3I Applications." In *Large Screen Projection Displays*, 760:6-10. SPIE Press, Bellingham, Washington.

Carey, J. M., ed. 1988. *Human Factors in Management Information Systems.* Ablex Publishing Corporation, Norwood, New Jersey.

Carlson, E. D., and W. Metz. 1980. "Integrating Dialog Management and Data Base Management." *Information Processing: Proceedings of the IFIP Congress*, Vol. 80, pp. 463-468. North-Holland Publishing Co., Amsterdam.

Carroll, J. M., and J. McKendree. 1987. "Interface Design Issues for Advice-Giving Expert Systems." *Communications of the ACM* 30(1):14-31.

Carroll, J. M., and S. A. Mazur. 1986. "LisaLearning." *Computer* 19(11):35-49.

Chao, B. P. 1986. *Design Guidelines for Human-Computer Dialogues.* SAND86-0259, Sandia National Laboratories, Albuquerque, New Mexico.

Chao, B. P. 1987. "Prototyping a Dialogue Interface: A Case Study." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, ed. G. Salvendy, pp. 357-363. Elsevier Science Publishers B.V., Amsterdam.

Chapnick, P. 1989. "Ushering in a New Era." *AI Expert*, Feb. 1989, p. 7-8, Miller Freeman, Inc.

Chimera, R. 1993. "Evaluation of Platform Independent User Interface Builders." (HCIL 93-04) Human-Computer Interaction Laboratory, University of Maryland, College Park, Maryland.

Christensen, C., E. A. Baggen, and H. L. Snyder. 1985. "Performance Measures and Subjective Evaluations for Two Color Displays." In *Proceedings of the Human Factors Society - 29th Annual Meeting*, pp. 1075-1078. Santa Monica, California.

Clarke, A. A. 1986. "A Three-Level Human-Computer Interface Model." *International Journal of Man-Machine Studies* 24:503-517.

Cowen, M. 1991. *A Comparison of Four Types of Feedback During Computer-Based Training (CBT).* NPRDC-TR-92-2, Navy Personnel Research and Development Center, San Diego, California.

Crolotte, A., J. Saleh, and A. Freedy. 1980. "Human Decision Processes in Command and Control of Marine Amphibious Brigade Operations," *IEEE*, pp. 1216-1220.

Cuff, R. N. 1980. "On Casual Users." *International Journal of Man-Machine Studies* 12(2):163-187.

Davies, S. P., A. J. Lambert, and J. M. Findlay. 1989. "The Effects of the Availability of Menu Information During Command Learning in a Word Processing Application." *Behavior and Information Technology* 8:135-144.

Decker, J. J., C. J. Dye, C. J. C. Lloyd, and H. L. Snyder. 1991. *The Effects of Display Failures and Symbol Rotation on Visual Search and Recognition Performance.* Technical Memorandum 4-91. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Decker, J. J., P. L. Kelly, K. Kurokawa, and H. L. Snyder. 1991. *The Effect of Character Size, Modulation, Polarity, and Font on Reading and Search Performance in Matrix-Addressable Displays*. Technical Memorandum 6-91. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Defense Intelligence Agency (DIA). 1990. "DIA Standard Military Graphics Symbols Manual" (DIAM 65-xx) (Draft). DIA, Washington, D.C.

Defense Intelligence Agency (DIA). 1989. *Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines*. DIA Document DDS-2600-6215-89, DIA, Washington, D.C.

Defense Intelligence Agency (DIA). 1983. *DIA Standard User Interface Style Guide for Compartmented Mode Workstations*. DIA Memorandum U-15, 284/DSE-3, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1994. *Technical Reference Model for Corporate Information Management, Version 2.0*. DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1993. *Department of Defense Technical Architecture Framework for Information Management, Volume 8. Human Computer Interface Style Guide, Version 3.0*. DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1992a. *Comparison of the IEEE Recommended Practice for Graphical User Interface Drivability with the DISA Center for Information Management Human Computer Interface Style Guide*. Technical Memorandum, DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1992b. *Human Computer Interface Style Guide, Version 1.0*. DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1992c. *Human Computer Interface Style Guide, Version 2.0*. DISA/CIM, Washington, D.C.

Diethelm, W., and M. Diethelm. 1984. *Signet Signal Symbol, Handbook of International Signs*. ABC Edition, Zurich.

Doll, T. J. 1991. "Electro-Optic/Infrared Technology and the Human-Machine Interface." In *Proceedings of the Human Factors Society - 35th Annual Meeting*, pp. 1500-1501. HFS, Santa Monica, California.

Dominessy, M. E. 1989. *A Literature Review and Assessment of Touch Interactive Devices*. Technical Memorandum 11-89. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Dreyfuss, H. 1984. *Sourcebook to International Graphic Symbols*. Van Nostrand Reinhold Company, Inc., New York.

Dumas, J. S. 1988. *Designing User Interfaces for Software*. Prentice Hall, Englewood Cliffs, New Jersey.

Durrett, H. J., ed. 1987. *Color and the Computer*. Academic Press, Inc., Boston.

Dye, C. J., and H. L. Snyder. 1991. *The Effects of Display Failures, Polarity, and Clutter on Visual Search for Symbols on Cartographic Images*. Technical Memorandum 9-91, U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Eberts, R. E. 1994. *User Interface Design*. Prentice Hall, Englewood Cliffs, New Jersey.

Egan, D. E. 1988. "Individual Differences in Human-Computer Interaction. "In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 543-568. North Holland: Elsevier Science Publishers, B.V., Amsterdam.

Ehrenreich, S. L. 1981. "Query Languages: Design Recommendations Derived from the Human Factors Literature." *Human Factors* 23(6):709-725.

Ehrhart, L. S. 1990. "New Approaches to Human-Computer Interaction Research and Design for Decision Aiding Systems." In *Proceedings of the 5th IEEE International Symposium on Intelligent Control*. IEEE Computer Society Press, Los Alamitas, California.

Eisen, H. A. 1990. "Iconer: A Tool for Evaluating Icons." *SIGCHI Bulletin* 21(3):23-25.

Elkerton, J. and R. C. Williges. 1989. "Dialogue Design for Intelligent Interfaces." *Intelligent Interfaces: Theory, Research, and Design*, eds. P. A. Hancock and M. H. Chignell.

Elkerton, J., R. C. Williges, J. A. Pittman, and J. Roach. 1982. "Strategies of Interactive File Search" in *Proceedings of the Human Factors Society*. HFS, Santa Monica, California.

Farooq, M. U., and W. D. Dominick. 1988. "A Survey of Formal Tools and Models for Developing User Interfaces." *International Journal of Man-Machine Studies* 29:479-496.

Feldman, M. B., and G. T. Rogers. 1982. "Toward the Design and Development of Style-Independent Interactive Systems." In *Human Factors in Computer Systems: Proceedings*, March 15-17, 1982, pp. 111-116. ACM, New York.

Fenchel, R. 1981. "An Integrated Approach to User Assistance." *ACM SIGSOC Bulletin 13* 2:98-104.

Fernandes, K. 1992. *User Interface Specifications for Navy Command and Control Systems, Version 1.1*. U.S. Department of the Navy; Naval Command, Control, and Ocean Surveillance Center, Research, Development, Test, and Evaluation Division, San Diego, California.

Fernandes, K., and K. E. Maracle. 1991. *Design Standards for the Command and Control Information Navigation and Training System (CINTS)*. Technical Note 1660, Naval Ocean Systems Center, San Diego, California.

Finke, D. J., and M. M. Lloyd. 1988. "Guidelines and Tools for Human-Decision Aid Interface Design and Evaluation." In *Proceedings of the Fifth Annual Workshop on Command and Control Decision Aiding*, pp. 170-193, September 1988.

FIPS - See National Institute of Standards and Technology (NIST)

Flynn, M. K. 1993. "Multimedia PC Gets Updated." *PC Magazine*. 12(13):30.

Flynn, R. R. 1987. *An Introduction to Information Science*. Marcel Dekker, Inc., New York.

Fowler, S. L., and V. R. Stanwick. 1995. *The GUI Style Guide*. AP Professional, Imprint of Academic Press, Inc., Cambridge, Massachusetts.

Fowler, C. J. H., L. A. Macaulay, and J. F. Fowler. 1985. "The Relationship Between Cognitive Style and Dialogue Style: An Explorative Study." In *People and Computers: Designing the Interface*, eds. P. Johnson and S. Cook, pp. 186-198. Cambridge University Press, Cambridge.

Frost, R. A. 1984. *Database Management Systems*. McGraw-Hill, New York.

Frutiger, A. 1980. *Type Sign Symbol*. ABC Verlag, Zurich.

Funk, H. L. 1989. "Information Display - An Overview and Trends." In *Proceedings of the Institution of Electrical and Electronics Engineers*, pp. 2-1 - 2-7. IEEE, Washington, D.C.

Ga Cote, R. 1992. "Code on the Move." *Byte Magazine*. Jul 92, pp.206-226.

Gaines, B. R., and M. L. G. Shaw. 1986. "Foundation of Dialog Engineering: The Development of Human-Computer Interaction." Part II. *International Journal of Man-Machine Studies* 24:101-123.

Gaines, B. R., and P. V. Facey. 1975. "Some Experience in Interactive System Development and Application." In *Proceedings of the IEEE* 63(6):894-911.

Galitz, W. O. 1994. *It's Time To Clean Your Windows*. John Wiley & Sons, Inc., New York.

Galitz, W. O. 1993. *User-Interface Screen Design*. John Wiley & Sons, Inc., New York.

Galitz, W. O. 1989. *Handbook of Screen Format Design*. QED Information Sciences, Inc., Wellesley, Massachusetts.

Galitz, W. O. 1984. *Humanizing Office Automation*. QED Information Sciences, Inc., Wellesley, Massachusetts.

Garner, K. H. 1990. "20 Rules for Arranging Text on a Screen." *CBT Directions*:13-16.

General Services Administration (GSA). 1991. *Managing Information Resources for Accessibility*. GSA, Clearinghouse on Computer Accommodation (COCA) of the Information Resources Management Service (IRMS), Washington D.C.

Gery, G. 1991. *Electronic Performance Support Systems*. Weingarten Publications, Boston, Massachusetts.

Getler, R. 1991. "The Case for Concurrent Authoring." *CBT Directions*:14-22.

Gilmore, W. E., D. I. Gertman, and H. S. Blackman. 1989. "The User-Computer Interface in Process Control." In *A Human Factors Engineering Handbook*, pp. 21-33. EG&G Idaho, Inc., Idaho Falls, Idaho.

Gittins, D. 1986. "Icon-Based Human-Computer Interaction." *International Journal of Man-Machine Studies* 24:519-543.

Glasser, J., and A. Rolland. 1989. "Visual Performance Evaluation for LCD Displays: Appropriate Methods for Measuring Luminance and Contrast." *Proceedings of Society for Information Display*, 1077:9-20. SID, New York.

Gold, R. S., and A. G. Ledebuhr. 1985. "Full Color Liquid Crystal Light Valve Projector." In *Advanced in Display Technology V*, 526:51-58. SPIE Press, Bellingham, Washington.

Goldberg, A., and D. Robson. 1985. *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, Reading, Massachusetts.

Goode, W. F. 1991. "Status of Electronic Displays." Seminar for the Society of Information Display, SID, New York.

Gordon, S. E. 1988. "The Human Factor in Expert Systems," *AI Expert*, pp. 55-59.

Grabinger, R. S., and D. Amedeo. 1988. "CRT Text Layout: Perceptions of Viewers." *Computers in Human Behavior* 4:189-205.

Green, P., and W. T. Burgess. 1980. *Debugging a Symbol Set for Identifying Dsiplays: Production and Screening Studies*. Highway Safety Research Institute, University of Michigan, Ann Arbor, Michigan.

Greenberg, S., and I. H. Witten. 1985. "Adaptive Personalized Interfaces -- A Question of Viability." *Behaviour and Information Technology* 4(1):31-45.

Greitzer, F. 1987. Comments on draft NASA Guidelines Document: *User/Computer Interaction Section*. Memo on 1, 22, 1987.

Grether, W. F., and C. A. Baker. 1972. *Human Engineering Guide to Equipment Design*, eds., H. P. Van Cott and R. C. Kinkade. GPO, Washington, DC.

Grill, E. 1990. *Relational Databases: A Methodical Guide for Practical Design and Implementation.* Ellis Harwood, New York.

GSA - See General Services Administration

Gunderson, J., G. Gruetzmacher, and N. Swanson. 1991. "Legibility of Seven Segment Numeric LED Displays: Comparisons of Two Fonts at Various Distances." In *Proceedings of the Human Factors Society - 35th Annual Meeting*, pp. 491-495. HFS, Santa Monica, California.

Hagan, T. 1992. "GUI Tools Boost Portability." *Open Systems Today* 28 Sep 92, pp.39-44.

Hamel, C. J., and S. L. Clark. 1986. *CAI Evaluation for the Design of Computer-Aided Instruction.* Technical Report NTSC TR86-002 (DTIC No. AD-A172383). Naval Training Systems Center, Orlando, Florida.

Hannigan, S., and V. Herring. 1987. "Human Factors in Office Product Design -European Practice." In *Proceedings of the Second International Conference on Human-Computer Interaction*, Conference Title "Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems," ed. G. Salvendy, pp. 226-232. Elsevier Science Publishers B.V., Amsterdam.

Hansen, G. W., and J. V. Hansen. 1992. *Database Management and Design.* Prentice Hall, Englewood Cliffs, New Jersey.

Hansen, W. J. 1971. "User Engineering Principles for Interactive Systems." *Proceedings Fall Joint Computer Conference*, pp. 523-543. AFIPS Press, Montvale, New Jersey.

Harper, E. 1992. "WinLogin Offers Control Over Configuration Files." *LAN Times* 9(21):92.

Harrell, T. H. 1987. "Designing User-Computer Dialogues: Basic Principles and Guidelines." In *Computerized Psychological Testing: Current Issues and Future Directions*, Honaker, L. M., Chair. Symposium conducted at the American Psychological Association, New York.

Harter, S. P. 1986. *On-line Information Retrieval.* Academic Press, Orlando, Florida.

Hawkridge, D. G. 1988. *Computers in Company Training.* Croom Helm, London.

Heines, J. M. 1984. *Screen Design Strategies for Computer-Assisted Instruction.* Digital Press, Bedford, Massachusetts.

Helander, M., ed. 1988. *Handbook of Human-Computer Interaction.* Elsevier Science Publishers B.V., Amsterdam.

Henry, T. R., and S. E. Hudson. 1990. "Multidimensional Icons." *ACM Transactions on Graphics* 9(1):133-137.

Hershman, R. L., R. T. Kelly, and H. G. Miller. 1979. *User Performance With a Natural Language Query System for Command Control.* Navy Personnel Research and Development Center, San Diego, California.

HFS - See Human Factors Society

Hildreth, C. R. 1982. *Online Public Access Catalogs: The User Interface.* Online Computer Library Center, Inc. Dublin, Ohio.

Hix, D. 1991. "An Evaluation Procedure For User Interface Development Tools, Version 2.0." Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Hix, D., and R. S. Schulman. 1991. "Human-Computer Interface Development Tools: A Methodology For Their Evaluation." *Communications of the ACM* 34(3):75-87.

Hoadley, E. D. 1990. "Investigating the Effects of Color." *Communications of the ACM* 33(2):120-127.

Holtzman, S. 1989. *Intelligent Decision Systems.* Addison-Wesley, Reading, Massachusetts.

Horton, W. 1994. *The Icon Book, Visual Symbols for Computer Systems and Documentation.* John Wiley & Sons, New York.

Horton, W. K. 1990. *Designing and Writing Online Documentation - Help Files to Hypertext.* John Wiley & Sons, Inc. New York.

Human Factors Society, Inc. (HFS). 1988. *American National Standard for Human Factors Engineering of Visual Display Terminal Workstations.* ANSI/HFS 100-1988, HFS, Santa Monica, California.

Humphrey, S. M., and B. J. Melloni. 1986. *Databases: A Primer for Retrieving Information by Computer.* Prentice Hall, Englewood Cliffs, New Jersey.

Hunter, M. W. 1988. "CRT Anti-glare Treatments, Image Quality, and Human Performance." Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Hurd, J. C. 1983. "Writing Online Help." In *Proceedings of 30th International Technical Communication Conference*, pp. 151-154. Society for Technical Communication, Washington, D.C.

Hurley, A. F., and L. I. Hoffberg. 1991. *Evaluation of Six Large-Screen Displays.* Technical Report No. 29399, The MITRE Corp., Bedford, Massachusetts.

Hussein, B. 1989. "DSS Products Evaluation: An Integrated Framework." *Journal of Systems Management*, pp. 27-34, November 1989.

Hyland, R. M., and P. R. Boivin. 1987. *Performance Characteristics of Infrared Touch Screens for Typical Workstation Applications.* Technical Report No. NSU 8093, Defense Technical Information Center, Alexandria, Virginia.

IBM. 1984. *Human Factors Workstations with Visual Displays,* Third Edition. San Jose, California.

Inmon, W. H. 1981. *Effective Data Base Design.* Prentice Hall, Englewood Cliffs, New Jersey.

Innocent, P. R. 1982. "Towards Self-Adaptive Interface Systems." *International Journal of Man-Machine Studies* 16:287-299.

Institute of Electrical and Electronics Engineers (IEEE). 1993a. "Draft Standard for Information Technology-Uniform Application Program Interface-Graphical User Interfaces, "IEEE P1201.1, Draft 7. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

Institute of Electrical and Electronics Engineers (IEEE). 1993b. "IEEE Recommended Practice for Graphical User Interface Drivability," IEEE P1201.2, Draft 2. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

Institute of Electrical and Electronics Engineers (IEEE). 1993c. *Standards for Information Technology -- X Window System -- Modular Toolkit Environment (MTE),* IEEE P1295. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

Institute of Electrical and Electronics Engineers (IEEE). 1992. "Draft Standard for Information Technology-Uniform Application Program Interface-Graphical User Interfaces," IEEE P1201.1, Draft 3.1. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

International Organization for Standardization (ISO). 1991. "Part 3 Flat Panel Addendum, Additional Notes for Reviewers." In "ISO 9241, Working Draft of ISO Committee 159 SC 4 Working Group 2." ISO 9241, ISO, Washington, D.C.

Jarvenpaa, S. L., and G. W. Dickson. 1988. "Graphics and Managerial Decision-Making: Research Based Guidelines." *Communications of the ACM* 31(6):764-775.

Jorna, R. 1988. "Chapter 10. A Comparison of Presentation and Representation: Linguistic and Pictorial." In *Human-Computer Interaction: Psychonomic Aspects,* eds. G. C. van der Veer and G. Mulder, pp. 172-185. Springer-Verlag, Berlin Heidelberg.

Karon, P. 1992. "Cross-Platform Tools Appeal to Developers." *InfoWorld* 17 Aug 92, pp. S74-S75.

Katzeff, C. 1986. "Dealing With a Database Query Language in a New Situation." *International Journal of Man-Machine Studies* 5(1):1-17.

Katzeff, C. 1988. "The Effect of Different Conceptual Models Upon Reasoning in a Database Query Writing Task." *International Journal of Man-Machine Studies* 29(1):37-62.

Kawamura, T., H. Kawamura, K. Kobara, A. Saitoh, and Y. Endo. 1991. "Anti-reflection Coating for Inner Surface of CRT Faceplate." In *Society for Information Display 91 Digest*, pp. 49-52. SID, New York.

Kearsley, G. 1988. *Online Help Systems: Design and Implementation*. Ablex Publishing, Norwood, New Jersey.

Kelley, J. F. 1984. "An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications." *ACM Transactions on Office Information Systems* 2(1):26-41.

Kelster, R. S., and G. R. Galloway. 1983. "Making Software User Friendly: An Assessment of Data Entry Performance." *Proceedings of the Human Factors Society*. HFS, Santa Monica, California.

Kieras, D. E., and S. Boviar. 1984. "The Role of a Mental Model in Learning to Operate a Device." *Cognitive Science* 8:255-273.

Klien, G. A., and D. MacGregor. 1988. "Knowledge Elicitation of Recognition-Primed Decision-Making." *Technical Report 799*. U.S. Army Research Institute Field Office, Fort Leavenworth, Kansas.

Kobara, S. 1991. *Visual Design with OSF/Motif*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Korth, H. F., A. Silbershatz. 1986. *Database System Concepts*. McGraw-Hill, New York.

Krauşe, J. 1980. "Natural Language Access to Information Systems: An Evaluation Study of Its Acceptance by End Users." *Information Systems* 5(4):297-318.

Kubota, H., T. Marushige, C. M. Gomes, and S. Kobayashi. 1988. "Legibility of Multiplexed Dot-Matrix LCDS: The Effect of Surface Reflection." In *Proceedings of Society for Information Display (SID)*, 29(3):213-216. SID, New York.

Kubota, H., T. Marushige, T. Takabayashi, and S. Kobayashi. 1986. "LCD Legibility." In *Society for Information Display 88 Digest*, pp. 157-160. SID, New York.

Kuwayama, Y. 1973a. *Trademarks & Symbols, Volume 1: Alphabetical Designs*. Van Nostrand Reinhold Company, New York.

Kuwayama, Y. 1973b. *Trademarks & Symbols, Volume 2: Symbolical Designs*. Van Nostrand Reinhold Company, New York.

Langen, M., B. Thull, T. Schecke, G. Rau, and G. Kalff. 1989. "Prototyping Methods and Tools for the Human-Computer Interface Design of a Knowledge-Based System." *Designing* *and Using Human-Computer Interfaces and Knowledge-Based System: Proceedings of the Third International Conference on Human-Computer Interaction*, Vol. II, eds. G. Salvendy and M. J. Smith, pp. 861-868. September 18-22, 1989, Boston. Elsevier Science Publishers B.V., Amsterdam.

Lansdale, M. W. 1988. "On the Memorability of Icons in an Information Retrieval Task." *Behaviour and Information Technology* 7(2):131-151.

Lanzetta, T. M., and N. D. Lubart. 1988. "Comparing the Readability of Display Technologies: Paper, CRT, and LCD." In *Society for Information Display 88 Digest*, pp. 336-339. SID, New York.

Larson, H. T. 1989. "Large Automated Flat Situation Display for the Commander." *Maintaining Start-of-the Art in ACCS*:3-44. Army Science Board Summer Study.

Laurel, B., ed. 1990. *The Art of Human Computer Interface Design*. Addison-Wesley, Reading, Massachusetts.

Laverson, A., K. Norman, and B. Shneiderman. 1987. "An Evaluation of Jump-Ahead Techniques in Menu Selection." *Behavior and Information Technology* 6(2):97-108.

Laycock, J. 1985. "The Legibility of Passive Displays". *Proceedings of Society for Information Display*, 26(2):89-93. SID, New York.

LeMay, M. 1988. "Why Some Decision Aids Work and Others Do Not." *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 227-229.

Lerner, N. D., and B. L. Collins. 1980. *The Assessment of Safety Symbol Understandability by Different Testing Methods*. U.S. Department of Commerce, Washington, D. C.

Lewis, H. V., and J. J. Fallesen. 1989. *Human Factors Guidelines for Command and Control Systems: Battlefield and Decision Graphics Guidelines*. Research Project 89-01. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Lickteig, C. W. 1989. *Design Guidelines and Functional Specifications for Simulation of the Battlefield Management Systems (BMS) User Interface*. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Lloyd, C. J. C., J. J. Decker, and H. L. Snyder. 1991. *The Effects of Line and Cell Failures on Reading and Search Performance Using Matrix-Addressable Displays*. Technical Memorandum 7-91. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Lloyd, C. J. C., J. J. Decker, K. Kurokawa, and H. L. Snyder. 1988. "Effects of Line and Cell Failures on Reading and Search Performance Using Matrix-Addressable Displays." In *Society for Information Display 88 Digest*, pp.344-346. SID, New York.

Lochovsky, F. H, and D. C. Tsichritzis. 1984. "Querying External Databases." In *Human Factors and Interactive Computer Systems*, ed. Y. Vassiliou. Ablex Publishing Corporation, Norwood, New Jersey.

Lodding, K. N. 1983. "Iconic Interfacing." *IEEE Computer Graphics and Applications* 3(2):11-20.

Luther, A. C. 1992. *Designing Interactive Multimedia*. Bantam Books, New York.

Lysaght, R. J., R. Harris, and W. Kelly. 1988. "Artificial Intelligence for Command and Control." *Technical Report 2122*. U.S. Army Communications and Electronics Command, Fort Monmouth, New Jersey.

Ma, P., F. H. Murphy, and E. A. Stohr. 1989. "A Graphics Interface for Linear Programming," *Communications of the ACM* 32:996-1012.

MacGregor, J. M., and E.S. Lee. 1988. "A Feature Matching Approach to the Retrieval of Graphical Information." *Behavior and Information Technology* 7(4):457-465.

MacGregor, R. C. , K. F. King, and R. J. Clarke. 1988. "Individualising the Man-Machine Interface." In *Ergonomics of Hybrid Automated Systems, Proceedings of the First International Conference on Ergonomics of Advanced Manufacturing and Hybrid Automated Systems*, eds. W. Karwowski, H. R. Parsaei, and M. R. Wilhelm, pp. 275-281. Louisville, Kentucky.

Magnavox, Inc. 1985. *Appendix B - Application of Data Processing Networking Techniques for Army Command and Control Systems (ACCS) Task 1 Subtask 4.* Report No. 11-89 on Contract No. DAAK11-84-D-0006, Magnavox, Inc., Fort Wayne, Indiana.

Main, R. and D. Paulson. 1988. *Guidelines for the Development of Military Training Decision Aids*. NPRDC TR 88-16. Navy Personnel Research and Development Center, San Diego, California.

Mallary, T. C. 1985. "Design of the Human-Computer Interface for a Computer Aided Design Tool for the Normalization of Relations." Master's Thesis, Air Force Institute of Technology, Wright Air Force Base, Ohio.

Marcus, A., N. Smilonich, and L. Thompson. 1995. *The Cross-GUI Handbook*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Marcus, A. 1992. *Graphic Design for Electronic Documents and User Interfaces*. ACM Press, New York.

Marcus, A. 1984. "Corporate Identity for Iconic Interface Design: The Graphic Design Perspective." *IEEE Computer Graphics and Applications (CG&A)* 4(12):24-32.

Marcus, A. 1979. *Managing Facts and Concepts.* National Endowment for the Arts, Washington, D.C.

Martin, J. 1983. *Managing the Data-Base Environment.* Prentice Hall, Englewood Cliffs, New Jersey.

Masiaszek, L. A. 1990. *Database Design and Implementation.* Prentice Hall, New York.

Matthews, M. L. 1987. "The Influence of Color on CRT Reading Performance and Subjective Comfort Under Operational Conditions." *Applied Ergonomics* 18.4:323-328.

McCann, C. 1988. Final Report from "Research Study Group 12 (RSG.12) on Computer-Human Interaction in Command and Control." Document AC/243 (Panel 8/RSG.12) D/7, DCIEM, Downsview, Ontario.

McCann, P. H. 1983. *Methods for Improving the User-Computer Interface.* Report No. NPRDC TR 83-29, Navy Personnel Research and Development Center, San Diego, California.

McKeown, P. E., J. J. Fallesen, M. S. Perkins, and C. G. Ross. 1991. *Operations Planning Tools (OPT) Functional Description.* U.S. Army Research Institute Product 91-09 (AD-A235 665), Fort Leavenworth, Kansas.

McNeese, M. D., and L. Katz. 1986. "Legibility Evaluation of a Large-Screen Display System Under Medium Ambient Illumination." In *Society for Information Display 86 Digest*, pp. 142-145. SID, New York.

Meyer, A. 1992. "Developing a Portable C++ GUI Class Library." *Dr. Dobbs Journal,* Nov 92, pp. 102-109.

Microsoft Corporation. 1992. *The Windows^{TM} Interface: An Application Design Guide.* Microsoft Press, Redmond, Washington.

Microsoft Corporation. 1991a. *Microsoft® C/C++ ver 7.0 Tutorial.* Doc No. LN24772-1191, Microsoft Press, Redmond, Washington.

Microsoft Corporation. 1991b. *Microsoft® Windows^{TM} Multimedia Authoring and Tools Guide.* Microsoft Press, Redmond, Washington.

Milgram, P., D. Drascic, and J. J. Grodski. 1991. "Enhancement of 3-D Video Displays by Means of Superimposed Stereo-Graphics." In *Proceedings of the Human Factors Society - 35th Annual Meeting*, pp. 1457-1461. HFS, Santa Monica, California.

Milheim, W.D., and B. L. Martin. 1991. "Theoretical Basis for the Use of Learner Control: Three Different Perspectives." *Journal of Computer-Based Instruction* 18(3):99-105.

Miller, P. J. 1991. *NMR Display Prototype Experiment Results.* Technical Report No. AD-A239 131, Defense Technical Information Center, Alexandria, Virginia.

Minasi, M. 1994. *Secrets of Effective GUI Design.* SYBEX, Inc., Alameda, California.

Minasi, M. 1990. "Bayes and Simple Expert Systems." *AI Expert*, pp. 13-15.

Mitchell, D. K., and K. P. Kysor. 1992. *A Preliminary Evaluation of the Prototype Tactical Computerized Interactive Display*. Technical Memorandum 2-92. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Mittal, S. 1985. "Knowledge Acquisition from Multiple Experts. *AI Magazine*, pp. 32-36.

Miyashita, T., and T. Uchida. 1990. "Cause of Fatigue and its Improvement in Stereoscopic Displays." *Proceedings of Society for Information Display*, 31(3):249-254. SID, New York.

Moran, T. P. 1981. "The Command Language Grammar: A Representation for the User of Interactive Computer Systems." *International Journal of Man-Machine Studies* 15:3-50.

Morse, R. S. 1985. "Glare Filter Preference: Influence of Subjective and Objective Indices of Glare, Sharpness, Brightness, Contrast and Color." In *Proceeding of the Human Factors Society - 29th Annual Meeting*, pp. 782-786. HFS, Santa Monica, California.

Muracka, T., M. Kawamura, and H. Uesako. 1989. "Readability on the Positive Type Liquid Crystal Display Devices with Multinumerals Influenced by the Irradiation Illuminances." In *Work With Computers: Organizational, Management, Stress and Health Aspects*, eds. M. J. Smith and G. Salvendy, pp. 542-548. Elsevier Science Publishers B.V., Amsterdam.

Murphy, T. 1993. "Looking at the World Through Cheap Sunglasses." *Computer Language* 10(2):63-85.

Muter, P., and C. Mayson. 1986. "The Role of Graphics in Item Selection from Menus." *Behaviour and Information Technology* 5(1):89-95.

National Institute of Standards and Technology (NIST). 1993. "User Interface Component of Applications Portability Profile." *Federal Information Processing Standard (FIPS) 158-1*. NIST, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1991a. *Application Portability Profile (APP) The U.S. Government's Open Systems Environment Profile OSE/1, Version 1.0* (FIPS Pub 151-1), NIST Special Report 500-187, National Institute of Standards and Technology, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1991b. *Government Open Systems Interconnection Profile (GOSIP), Version 1.0* (FIPS Pub 146-1), National Institute of Standards and Technology, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1990. *POSIX, Portable Operating System Interface for Computer Environments (IEEE 1003.1-1988)* (FIPS Pub 151-1), National Institute of Standards and Technology, Gaithersburg, Maryland.

Nes, F. 1986. "Space, Color, and Typography on Visual Display Terminals." *Behavior and Information Technology* 5(2):99-118.

Neurath, O. 1980. *International Picture Language/Internationale Bildersprache.* University of Reading, England.

Neuron Data, Inc. 1992a. *Open Interface Technical Overview.* Neuron Data, Inc., Palo Alto, California.

Neuron Data, Inc. 1992b. *Open Interface Programmer's Guide, Version 2.0.* PN Man-30-400-02, Neuron Data, Inc, Palo Alto, California.

Nickerson, R. S. 1986. *Using Computers: The Human Factors of Information Systems.* The MIT Press, Cambridge, Massachusetts.

Nicol, A. 1990. "Interface for Learning: What Do Good Teachers Know That We Don't?" In *The Art of Human-Computer Interface Design*, ed. B. Laurel, pp. 113-122. Addison-Wesley, Reading, Massachusetts.

Nielsen, J. 1987. "A User Interface Case Study of the Macintosh." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems.* ed. G. Salvendy, pp. 241-248. Elsevier Science Publishers B.V., Amsterdam.

Nielsen, J. 1990. *Hypertext and Hypermedia.* Academic Press, Inc., Boston.

NIST - see National Institute of Standards and Technology

Nolan, P. R. 1989. "Designing Screen Icons: Ranking and Matching Studies." In *Proceedings of the Human Factors Society 33rd Annual Meeting, Volume 1*, pp. 380-384. Human Factors Society, Santa Monica, California.

Norman, K. L. 1991. *The Psychology of Menu Selection: Designing Cognitive Control of the Human/Computer Interface.* Ablex Publishing Corporation, Norwood, New Jersey.

Norman, K. L., and J. P. Chin. 1988. "The Effect of Tree Structure on Search in a Hierarchical Menu Selection System." *Behavior and Information Technology* 7(1):51-65.

North Atlantic Treaty Organization (NATO). 1990. *North Atlantic Treaty Organization Standardization Agreement 2019, Military Symbols for Land Based Systems.* U.S. Navy, Washington, D.C.

O'Keefe, R. M. 1989. "The Evaluation of Decision-Aiding Systems: Guidelines and Methods." *Information and Management* 17:217-226.

O'Malley, C., P. Smolensky, L. Bannon, E. Conway, J. Graham, J. Sokolov, and M. L. Montry. 1983. "A Proposal for User-Centered System Documentation." In HMI Project. *User-Centered System Design: Papers for the CHI '83 Conference on Human Factors in Computer Systems*, p.

4-8. DTIC No. AD 136131. (Technical Report). Office of Naval Research/Personnel and Training, Arlington, Virginia.

Ogden, W. C., and S. R. Brooks. 1983. "Query Languages for the Casual User: Exploring the Middle Ground between Formal and Natural Languages." In *Human Factors in Computing Systems: Proceedings of the CHI '83 Conference*, ed. A. Janda. North Holland Publishing Company, Amsterdam.

Olsen, L. A., and T. N. Huckin. 1991. *Technical Writing and Professional Communication (2nd ed.)*. McGraw-Hill, New York.

Olson, J. M. 1987. "Color and the Computer in Cartography." In *Color and the Computer*, ed. H. J. Durrett, pp. 205-219. Academic Press, Inc., San Diego, California.

Open Software Foundation (OSF). 1992. *OSF/Motif™ Style Guide*, Revision 1.2. Prentice Hall, Englewood Cliffs, New Jersey.

Osborn, P. B., and W. H. Zickefoose. 1990. "Building Expert Systems From the Ground Up." *AI Expert*, pp. 28-35.

OSF - see Open Software Foundation

Otte, F. H. 1982. "Consistent User Interface." In *Human Factors and Interactive Computer Systems*, pp. 261-275. Ablex, Norwood, New Jersey.

Owen, D. 1987. "Direct Manipulation and Procedural Reasoning." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, ed. G. Salvendy, pp. 349-356. Elsevier Science Publishers, B.V., Amsterdam.

Paap,.K. R., and R. J. Roske-Hofstrand. 1988. "Design of Menus." In *Handbook of Human-Computer Interaction*, ed. M. Helander, Elsevier Science Publishers, B.V., Amsterdam.

Paap, K. R., and R. J. Roske-Hofstrand. 1986. "The Optimal Number of Menu Options Per Panel." *Human Factors* 28(4):377-385.

Parker, S. P., ed. 1989. *McGraw-Hill Dictionary of Scientific and Technical Terms.* 4th ed. McGraw-Hill, New York.

Parkinson, S. R., M. D. Hill, N. Sisson, and C. Viera. 1988. "Effects of Breadth, Depth, and Number of Responses on Computer Menu Search Performance." *International Journal of Man-Machine Studies* 28:683-692.

Parrish, R. N., J. L. Gates, and S. J. Munger. 1981. *Design Guidelines and Criteria for User/Operator Transactions with Battlefield Automated Systems Volume IV: Provisional Guidelines and Criteria.* Technical Report 537, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Parsaye, K., M. Chignell, K. Setrag, and H. Wong. 1990. "Intelligent Databases." *AI Expert*, Mar. 1990, pp. 38-47, Miller Freeman, Inc.

Patricia Seybold's Office Computing Group. 1989. "Patricia Seybold's Office Computing Report." April 1989. 12(4):1(9).

Pavard, B. 1987. "Design of Graphic Dialogue Without Syntactic Constraints." In *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, ed. G. Salvendy, pp. 349-356. Elsevier Science Publishers B.V., Amsterdam.

Payne, S. J. 1983. "Readability of Liquid Crystal Displays: A Response Surface." *Human Factors Society, Inc.*, 25(2):185-190. Santa Monica, California.

Pejtersen, A. M., and L. P. Goodstein. 1990. "Beyond the Desk Top Metaphor: Information Retrieval With an Icon-Based Interface." *Visualization in Human-Computer Interaction, Interdisciplinary Workshop in Informatics and Psychology*. Springer-Verlag, Berlin.

Petrun, C. J., W. Hernon, and M. MacDonald. 1985. "An Examination of the Relative Legibility of Text on One Line Vacuum Florescent and Liquid Crystal Displays." In *Proceedings of the Human Factors Society - 29th Annual Meeting*, pp.1122-1124. HFS, Santa Monica, California.

Pew, R. W. 1988. "Human Factors Issues in Expert Systems." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 931-940. Elsevier Science Publishers B.V., Amsterdam.

Pleshko, P. 1991. "AC Electroluminescent Display Technology: Challenges and Potential." *Proceedings of Society for Information Display*, 32(2):106-108. SID, New York.

Rancourt, J., W. Grenawalt, M. W. Hunter, and H. L. Snyder. 1986. "Quantitative Evaluation of the Effect of an Antireflection Filter." In *Society for Information Display 86 Digest*, pp. 420-423. SID, New York.

Raum, H. G. 1988. *Design and Implementation of an Expert User Interface for the Computer Aided Prototyping System*. Thesis for U.S. Naval Postgraduate School, Monterey, California.

Raymond, E. S., ed. 1991. *The New Hacker's Dictionary*. The MIT Press, Cambridge, Massachusetts.

Reedy, A.E., D.L. Smith, and W.G. Bail. 1992. "Software Architecture Framework (Draft)," Defense Systems Information Agency/Center for Information Management, McLean, Virginia.

Reger, J. J., H. L. Snyder, W. W. Farley. 1989. "Legibility of Emissive and Non-Emissive Displays Under Florescent and Daylight Illumination." In *Society for Information Display 89 Digest*, pp. 364-367. SID, New York.

Reinhart, W. F. 1990. "Effects of Depth Cues on Depth Judgments Using a Field-Sequential Stereoscopic CRT Display." Unpublished Doctoral Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Relles, N. 1981. "A User Interface for On-line Assistance." In *Proceedings of the Fifth International Conference on Software Engineering*, pp. 400-408. Institute for Electrical and Electronics Engineers (IEEE), New York.

Relles, N., and L. A. Price. 1981. "A User Interface for On-line Assistance." In *Proceedings of the Fifth International Conference on Software Engineering*. Institute for Electrical and Electronics Engineers (IEEE), New York.

Rich, E. 1983. "Users Are Individuals: Individualizing User Models." *International Journal on Man-Machine Studies* 18:199-214.

Riedel, S. L. 1988. "User Acceptance and Field Implementation of Decision Support Systems." *Research Report 1477*. U.S. Army Research Institute Field Office, Fort Leavenworth, Kansas.

Ripley, G. D. 1989. "DVI--A Distal Multimedia Technology". *Communications of the ACM* 32:811-822.

Roach, J., R. Hartson, R. W. Ehrich, T. Yunten, and D. H. Johnson. 1982. "DMS: A Comprehensive System for Managing Human-Computer Dialogue." In *Human Factors in Computer Systems: Proceedings* March 15-17, 1982, Gaithersburg, Maryland.

Rockley, A. 1987. "Online Documentation: From Proposal to Finished Product." In *Proceedings of 34th International Technical Communication Conference*, ATA 58-61. Society for Technical Communication, Washington, D.C.

Rogers, Y. 1989. "Icons at the Interface: Their Usefulness." *Interacting With Computers - The Interdisciplinary Journal of Human-Computer Interaction* 1(1):105-117.

Rosch, W. L. 1994. *The Winn L. Rosch Hardware Bible*, 3rd Edition. Brady Publishing, Indianapolis, Indiana.

Rosenborg, V. 1993. *A Guide To Multimedia*. New Riders Publishing, Carmel, Indiana.

Rosenstein, M., and L. Weitzman. 1990. "The HITS Icon Editor - the Specification of Graphic Behavior Without Coding." *IEEE*, 0073-1129/90/0523:523-529.

Roth, J.L., J. A. Fitzpatrick, R. E. Warm, and J. L. Ditzian. 1988. Implementing *Embedded Training (ET): Volume 5 of 10: Designing the ET Component*. ARI Research Product No. 88-28, NTIS No. AD-A205 697. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Rupp, B. A. 1981. "Visual Display Standards: A Review of Issues." *Proceedings of Society for Information Display*, 22(1):63-72. SID, New York.

Salvendy, G. 1987. *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*. Elsevier Science Publishers B.V., Amsterdam.

Schauer, U. 1983. "The Integrated Data Analysis and Management System - A Generator for Enduser Systems." *Enduser Systems and Their Human Factors* eds. A. Blaser and M. Zoeppriz, pp. 30-61. Springer-Verlag, Berlin.

Schmitz, J. D., G. D. Armstrong, and J. D. Little. 1990. "Cover Story - Automated News Finding in Marketing." *Interfaces* 20(6):29-38.

Schur, S. 1988. "The Intelligent Database." *AI Expert*, Jan. 1988, pp. 26-34, Miller Freeman, Inc.

Schwartz, J. P. 1983. "Lack of Guidance for Decision Aid Interface Design." *SIGCHI Bulletin* 15:13-17.

Sellen, A., and A. Nicol. 1990. "Building User-Centered On-line Help." In *The Art of Human-Computer Interface Design*, ed. B. Laurel, pp. 143-153. Addison-Wesley, Reading, Massachusetts.

Seyer, P. 1991. *Understanding Hypertext Concepts and Applications*. Windcrest Books, Blue Ridge Summit, Pennsylvania.

Shaw, B., and M. McCauley. 1985. *Personal Computer Dialogue: A Human Engineering Data Base Supplement*. Technical Report No. AFAMRL-TR-85-013. U.S.A.F. Aerospace Medical Research Laboratory, San Antonio, Texas.

Shields, S. E., and W. P. Bleha. 1991. "Liquid Crystal Light Valves for Projection Displays." In *Society of Photoptic Instrumentation Engineers*, 1455:225-236. SPIE Press, Bellingham, Washington.

Shneiderman, B. 1992. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Second Edition. Addison-Wesley Publishing Company, Reading, Massachusetts.

Shneiderman, B. 1991. "Visual User Interfaces for Information Exploration," made available at the TAE Ninth User's Conference, New Carrollton, Maryland. CAR-TR-577, CS-TR-2748, Human-Computer Interaction Laboratory and Department of Computer Sciences, University of Maryland, College Park, Maryland.

Shneiderman, B. 1988. "We Can Design Better User Interfaces: A Review of Human-Computer Interaction Styles." *Ergonomics* 31(5):699-710.

Shneiderman, B. 1987. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, Massachusetts.

Shneiderman, B. 1984. "The Future of Interactive Systems and the Emergence of Direct Manipulation." In *Human Factors and Interactive Computer Systems*, ed. Y. Vassiliou, pp. 1-27. Ablex Publishing Corporation, Norwood, New Jersey.

Shneiderman, B. 1982a. "Designing Computer System Messages." *Communications of the ACM,* 25(9):604-605.

Shneiderman, B. 1982b. "Human Factors Experiments in Designing Interactive Systems." *Tutorial: End User Facilities in the 1980's,* ed. J. A. Larson, pp. 16-26. Proceedings of the IEEE Computer Society Sixth International Computer Software & Applications Conference, IEEE Computer Society Press, New York.

Shneiderman, B. 1978. "Improving the Human Factors Aspect of Database Interactions." *ACM Transactions on Database Systems* 3(4):417-439.

Shneiderman, B., and G. Kearsley. 1989. *Hypertext Hands-On! An Introduction to a New Way of Organizing and Accessing Information.* Addison-Wesley Publishing Company, Reading, Massachusetts.

Shurtleff, D. A., W. F. Wuersch, and J. G. Rogers. 1982. "Applications of Large-Screen Display Legibility Criteria." In *Society for Information Display 82 Digest*, pp. 202-203. SID, New York.

Shurtleff, D. A., W. F. Wuersch, and J. G. Rogers. 1981. "How to Make Large Screen Displays Legible." In *Proceedings of the Human Factors Society - 25th Annual Meeting*, pp. 149-153. HFS, Santa Monica, California.

Sidorsky, R. 1984. *Design Guidelines for User Transactions with Battlefield Automated Systems: Prototype for a Handbook.* DTIC AD-A153 231, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Silverstein, L. D., R. W. Monty, J. W. Huff, and K. L. Frost. 1987. *Image Quality and Visual Simulation of Color Matrix Displays.* Technical Paper No. 871789, The Engineering Society for Advancing Mobility Land, Sea, Air, and Space, Long Beach, California.

Sisson, N., S. R. Parkinson, and K. Snowberry. 1986. "Consideration of Menu Structure and Communication Rate for the Design of Computer Menu Displays." *International Journal of Man-Machine Studies* 25:479-489.

Slominski, S.E., and I. R. Young. 1988. *A User Friendly Design of an Interactive Prototype for the Maintenance and Monitoring of Civilian Training Records.* Unpublished Master's Thesis, Naval Postgraduate School, Monterey, California.

Smith, M. C., and L. E. Magee. 1980. "Tracing the Time Course of Picture-Word Processing." *Journal of Experimental Psychology: General* 109(4):373-392.

Smith, S. L., and J. N. Mosier. 1986. *Guidelines for Designing User Interface Software*. The MITRE Corp, Bedford, Massachusetts.

Smith, S. L., and J. N. Mosier. 1984. *A Design Evaluation Checklist for User Computer Interface Software*. The MITRE Corp., Bedford, Massachusetts.

Snyder, H. L. 1988. "Image Quality." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 437-474. Elsevier Science Publishers B.V., Amsterdam.

Snyder, H. L. 1987a. "The ANSI Human Factors Standard for Visual Display Terminal Workstation: A Process and Progress Report (Invited Address)." In *Society Information Display 87 Digest*, pp. 14-17. SID, New York.

Snyder, H. L. 1987b. "Counterintuitive Criteria for Visual Displays." In *Social, Ergonomic and Stress Aspects of Work with Computers*, eds. G. Salvendy, S. L. Sauter, nd J. J. Hurrell, Jr., pp. 145-156. Elsevier Science Publishers B. V., Amsterdam.

Snyder, H. L. 1984. "Ergonomics Database for Visual Displays and VDUs." In *Ergonomic Data for Equipment Design*, ed. H. Schmidtke, pp. 219-234. Plenum Press, New York.

Snyder, H. L. 1980. "Human Visual Performance and Flat Panel Display Image Quality." HFL-80-1/ONR-80-1, Human Factors Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Sormunen, E. 1987. "A Knowledge-Based Intermediary System for Information Retrieval," in *Knowledge Engineering Expert Systems and Information Retrieval*, ed. I. Wormell, pp. 59-73. Taylor Graham, London.

Spence, R., and M. Parr. 1991. "Cognitive Assessment of Alternatives." *Interacting With Computers* 3(3):270-282.

Stammers, R. B., and J. Hoffman. 1991. "Transfer Between Icon Sets and Ratings of Icon Concreteness and Appropriateness." In *Proceedings of the Human Factors Society 35th Annual Meeting*, Vol. 1, pp. 354-358. San Francisco.

Steele, C. A., J. Harrold, J. P. Stanton, and R. Daley. 1987. "Performance of Laser-Addressed Liquid Crystal Map Overlay Display." In *Society of Photoptic Instrumentation Engineers*, 760:70-73. SPIE Press, Bellingham, Washington.

Strategic Air Command (SAC). 1990. *Intelligence Data Handling System (IDHS) Man-Machine Interface (MMI) Style Guide, Version 1.0*. Washington, D.C.

Sun Microsystems, Inc. 1990. T*he Open Look Graphical User Interface Application Style Guidelines*. AT&T, New Jersey.

Suntola, T. 1989. "VLSI and Computer Peripherals: Thin Film EL-Display," *Proceedings of the Institution of Electrical and Electronics Engineers*, pp. 2-32/2-35. IEEE Computer Society Press, Washington, D.C.

Tannas, L. E., ed. 1985. *Flat-Panel Displays and CRTs*. Van Nostrand Reinhold, New York.

Tannenbaum, A. 1990. "Installing AI Tools Into Corporate Environments." AI Expert, pp. 54-59.

Tenopir, C., and G. Lundeen. 1988. *Managing Your Information*. Neal-Schuman Publishers, New York.

Thierauf, R. J. 1988. *User-Oriented Decision Support Systems*. Prentice Hall, Englewood Cliffs, New Jersey.

Thomas, J. C. 1983. "Psychological Issues in Data Base Management." In *Designing for Human-Computer Communications*, eds. M. E. Sime and M. J. Coombs, pp. 169-184. Academic Press, London.

Thomas, J.C., and J. M. Carroll. 1981. "Human Factors in Communication." IBM Systems Journal 20(2):236-263.

Thorell, L. G., and W. J. Smith. 1990. *Using Computer Color Effectively: An Illustrated Reference*. Prentice Hall, Englewood Cliffs, New Jersey.

TRW. 1990a. *The AWIS Common User/System Interface Style Guide*. AWIS Program Documentation.

TRW. 1990b. *UTACCS, Soldier-Machine Development Standards*. UTACCS Program Documentation.

Tullis, T. S. 1988. "Screen Design." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 377-411. Elsevier Science Publishers, B.V., Amsterdam.

Tullis, T. S. 1981. "An Evaluation of Alphanumeric, Graphic, and Color Information Displays." *Human Factors* 23(5):541-550.

Tzeng, O.C.S., N. T. Trung, and R. W. Rieber. 1990. "Cross-Cultural Comparisons on Psychosemantics of Icons and Graphics." *International Journal of Psychology* 25:77-97.

U.S. Department of the Army. 1987. *Map Reading and Land Navigation*. FM 21-26, Washington, DC.

U.S. Department of the Army. 1985a. *Authorized Abbreviations and Brevity Codes*. AR310-50, Army UPDATE Publications, Washington, D.C.

U.S. Department of the Army. 1985b. *Operational Terms and Symbols.* FM 101-5-1, U.S. Army Combined Arms Center, Fort Leavenworth, Kansas.

U.S. Department of the Army. 1984. *Human Engineering Guide to Equipment Design.* FM 21-26 (sec 11), eds. H. P. Van Cott and R. G. Kinkade. John Wiley & Sons, New York.

U.S. Department of Defense (DoD). 1994. *Common Warfighting Symbology, Version 1.* MIL-STD-2525, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1992a. *Department of Defense Human Computer Interface Style Guide* (Version 1.0), Defense Information Systems Agency/Center for Information Management, McLean, Virginia.

U.S. Department of Defense (DoD). 1992b. *Department of Defense Human Computer Interface Style Guide* (Version 2.0), Defense Information Systems Agency/Center for Information Management, McLean, Virginia.

U.S. Department of Defense (DoD). 1992c. "Standards-Based Architecture Planning Handbook, Draft Version 1.0." Produced by DMR Group, Inc. for DISA XI, Washington, D.C.

U.S. Department of Defense (DoD). 1992d. *Technical Reference Model for Information Management, Version 1.2*, Defense Information Systems Agency/Center for Information Management, McLean, Virginia.

U.S. Department of Defense (DoD). 1991a. *Department of Defense Intelligence Information Systems Style Guide*, Department of Defense Intelligence Information Systems (DODIIS) Management Board, Arlington, Virginia.

U.S. Department of Defense (DoD). 1991b. *DoD Directive 5000.1 Defense Acquisition*, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1991c. *DoD Instruction 5000.2. Defense Acquisition Management Policy and Procedures*, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1991d. *Air Crew Station Alerting Systems.* MIL-STD-411E, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1990. *Military Training Programs.* MIL-STD-1379D, Naval Sea Systems Command, SEA 55Z3.

U.S. Department of Defense (DoD). 1989a. *Department of Defense Dictionary of Military and Associated Terms.* DoD, Washington, D.C.

U.S. Department of Defense (DoD). 1989b. *Human Engineering Design Criteria for Military Systems, Equipment, and Facilities.* MIL-STD-1472D, U.S. Army Missile Command, Huntsville, Alabama.

U.S. Department of Defense (DoD). 1989c. *Human Engineering Guidelines for Management Information Systems.* DOD-HDBK-761A, DoD, Washington, D.C.

U.S. Department of Defense (DoD). 1988. *DoD Directive 7920.1. Life-Cycle Management of Automated Information Systems (AISs),* U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1984. *Legends for Use in Air Crew Stations and on Airborne Equipment.* MIL-STD-783D, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1981. *Abbreviations for Use on Drawings, Specification Standards, & in Technical Documents.* MIL-STD-12D, U.S. Department of Defense, Washington, D.C.

Ullman, J. D. 1988. *Principles of Database and Knowledge-Base Systems.* Computer Science Press, Rockville, Maryland.

UNIX System Laboratories, Inc. 1991. *Open Look Graphical User Interface User's Guide.* Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Urban, C. D. 1990. "Design and Evaluation of a Tactical Decision Aid." *Proceedings IEEE International Conference on Systems, Man, and Cybernetics.* DTIC No. AD-A232 001, IEEE, New York.

Valdes, R. 1992a. "Sizing Up Application Frameworks and Class Libraries." *Dr. Dobb's Journal* (Oct 92) pp. 18-27.

Valdes, R. 1992b. "Sizing Up GUI Toolkits." *Dr. Dobbs Journal* (Nov 92), pp. 18-26.

Van Cott, H. P., and R. G. Kinkade, eds. 1984. *Human Engineering Guide to Equipment Design.* John Wiley & Sons, New York.

Vassiliou, Y., ed. 1984. *Human Factors and Interactive Computer Systems.* Ablex Publishing Corporation, Norwood, New Jersey.

Vance, D. W. 1987. "Image Distortions in Video Projection." In *Large Screen Projection Displays*, 760:54-59. SPIE Press, Bellingham, Washington.

Vasta, J. A. 1985. *Understanding Data Base Management Systems.* Wadsworth Publishing Company, Belmont, California.

Vaughan, T. 1993. *Multimedia: Making It Work.* Osborne McGraw-Hill, Berkeley, California.

Veith, R. H. 1988. *Visual Information Systems: The Power of Graphics and Video.* G.K. Hall & Co., Boston.

Veron, H., D. A. Southard, J. R. Leger, and J. L. Conway. 1990. *3D Displays for Battle Management*. Technical Report No. AD-A223 142, Defense Technical Information Center, Alexandria, Virginia.

Veron, H. 1988. *A Resolution Measurement Technique for Large Screen Displays*. Technical Report No. M88-59, The MITRE Corp., Bedford, Massachusetts.

Verplank, W. L. 1988. "Graphic Challenges in Designing Object-Oriented User Interfaces." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 365-375. Elsevier Science Publishers B.V., Amsterdam.

Visix Software, Inc. 1993. *Galaxy Resource Builder User's Guide, Version 1.2*. Visix, Software, Inc., Reston, Virginia.

Visix Software, Inc. 1992. *Galaxy Application Environment Technical Description*. Visix Software, Inc., Reston, Virginia.

Waern, Y., and C. Rollenhagen. 1983. "Reading Text From Visual Display Units (VDUs)." In *International Journal of Man-Machine Studies* 18, Academic Press, London.

Walker, J. H. 1987. "Issues and Strategies for Online Documentation." *IEEE Transactions on Professional Communication, PC-30* (4):235-248.

Walrath, J. D. 1989. "Aiding the Decision-Maker: Perceptual and Cognitive Issues at the Human-Machine Interface." *Technical Note 15-89*. U.S. Army Human Engineering Laboratory (DTIC No. AD-A217 862), Aberdeen Proving Ground, Maryland.

Wehrer, W., and E. E. Mitchamore. 1992. "Technology/Marketing Case Study: Plug-and-play IR Touch Screens." *Information Display*, pp. 10-12. SID, New York.

Weinschenk, S., and S. C. Yeo. 1995. *Guidelines for Enterprise-Wide GUI Design*. John Wiley & Sons, New York.

Weingaertner, S. T., and A. H. Levis. 1988. "Evaluation of Decision Aiding in Submarine Emergency Decision-Making." IFAC Conference on the Analysis, Design and Evaluation of Man-Machine Systems, Ouhu, Finland, pp. 195-201.

Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman Publishers, Los Altos, California.

Wexelblat, R. L. 1989. "On Interface Requirements for Expert Systems." *AI Magazine* 10:66-78

Wiebe, A. F., D. Johnson, G. A. Harcey, and M. Teter. 1993. "The Task-Training Guideline: A Powerful Format for How-To Instructional Training Materials." *STC Technical Communication* 40(1):49-61.

Williams, K.E., C. J. Hamel, and L. B. Shrestha. 1987a. *An Evaluation of Characteristics Contributing Towards Ease of User-Computer Interface in a Computer-Aided Instruction Exercise*. Technical Report TR87-030, Naval Training Systems Center, Orlando, Florida.

Williams, K.E., C. J. Hamel, and L. B. Shrestha. 1987b. *CAI Evaluation Handbook: Guidelines for the User Interface Design for Computer-Aided Instruction*. Technical Report No. TR87-033, Naval Training Systems Center, Orlando, Florida.

Williams, R. D. 1990. "Volume 3D Display Technology." In *Stereographics*, 17th International Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, Special Interest Group on Graphics, Dallas, Texas.

Wilson, M., P. Barnard, and A. MacLean. 1990. An Investigation of the Learning of a Computer System. In *Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction*, ed. P. Falzon, pp. 151-172. Academic Press, London.

Witten, I. H., and S. Greenberg. 1985. "Adaptive Personalized Interfaces -- A Question of Viability." *Behavior and Information Technology* 4(1):31-45.

Wodaski, R. 1992. *Multimedia Madness!* Sams Publishing, Carmel, Indiana.

Wood, W. T., and S. K. Wood. 1987. "Icons in Everyday Life." In *Advances in Human Factors/Ergonomics, 10 A -- Proceedings of the Second International Conference on Human-Computer Interaction - Social, Ergonomic and Stress Aspects of Work with Computers*, eds. G. Salvendy, S. L. Sauter, and J. J. Hurrell, Jr., pp. 97-105, Honolulu, Hawaii.

Woodson, W. E., B. Tillman, and P. Tillman. 1992. *Human Factors Design Handbook: Information and Guidelines for the Design of Systems, Facilities, Equipment, and Products for Human Use*, Second Edition. McGraw-Hill, Inc., New York.

WordPerfect Corporation. 1991. *WordPerfect for Windows™, Reference*. WordPerfect Corporation, Orem, Utah.

Wormell, I., ed. 1987. *Knowledge Engineering Expert Systems and Information Retrieval.* Taylor Graham, London.

Wyatt, A. L. 1990. *Computer Professional's Dictionary*. Osborne McGraw-Hill, Berkeley, California.

XVT Software, Inc. 1992. *XVT Portability Toolkit Manual*. XVT Software, Inc., Boulder, Colorado.

Yamazaki, T., K. Kamijo, and S. Fukuzumi. 1990. "Quantitative Evaluation of Visual Fatigue Encountered in Viewing Stereoscopic 3D Displays: Near-Point Distance and Visual Evoked Potential Study." *Proceedings of Society for Information Displays*, 31(3):245-247. SID, New York.

Yeh, Y., and L. D. Silverstein. 1991. "Human Factors for Stereoscopic Color Displays." In *Society for Information Display 91 Digest,* pp. 826-829. SID, New York.

Yeh, Y., and L. D. Silverstein. 1989. "Using Electronics Stereoscopic Color Displays: Limits of Vision and Depth Discrimination." *Three-Dimensional Visualization and Display Technologies*, Proceedings of SPIE International Society for Optical Engineering, Vol. 1083, pp. 196-204. SPIE Press, Bellingham, Washington.

Zachary, W. W. 1988. "Decision Support Systems: Designing to Extend the Cognitive Limits." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 997-1030. Elsevier Science Publishers B.V., Amsterdam.

Ziegler, J. E., and K. P. Fähnrich. 1988. "Direct Manipulation." In *Handbook of Human-Computer Interaction*, ed. M. Helander, pp. 123-134. Elsevier Science Publishers B.V., Amsterdam.

Zmud, R. W. 1978. "Concepts, Theories and Techniques: An Empirical Investigation of the Dimension of the Concept of Information." *Decision Sciences* 9:187-195.